



Facultade de Informática

UNIVERSIDADE DA CORUÑA

TRABALLO FIN DE GRAO
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN ENXEÑARÍA DE COMPUTADORES

Dispositivo de prevención y alertas ciudadanas con tecnologías IoT

Estudiante: Iván Aguilar Gil

Dirección: Carlos J. Escudero Cascón

A Coruña, setembro de 2020.

A mis abuelos

Agradecimientos

Quiero agradecer a mi tutor por sus directrices y su paciencia.

A Alba por encaminarme hacia este proyecto y su continuo apoyo, al igual que su familia.

A mi familia por todos estos años a mi lado.

A todos los amigos que me acompañaron durante esta etapa de mi vida.

Resumen

Este proyecto consta de la creación de un dispositivo compacto, de bajo coste y alta eficiencia energética que utilizará tecnologías IoT. Este dispositivo IoT consistirá en el envío de una alerta a la nube que tras procesarla y obtener nuestra ubicación realizará una acción configurada con anterioridad por el propio usuario. Esta alerta se activa mediante una secuencia introducida por el usuario utilizando un pulsador. Este sistema permitirá una gran gama de acciones desde enviar la ubicación del dispositivo por Telegram hasta activar la alarma de seguridad de una casa si el dispositivo se encuentra dentro de la misma. Un ejemplo de uso sería:

- Un usuario configura en el sistema que en cuanto la secuencia se introduzca, se debe enviar la ubicación del dispositivo por SMS al teléfono de su pareja.
- Dado el momento, el usuario introduce la secuencia en el dispositivo y se activa la alerta.
- Esta es recibida por el servidor, que la procesa y obtiene la localización del dispositivo.
- Por último, el sistema envía por SMS la ubicación del dispositivo al teléfono establecido.

Además, en el proyecto, se analizan las tecnologías inalámbricas más conocidas y expandidas que se utilizan en IoT. Con esto, se realiza un análisis, con el fin de determinar la razón del por qué se ha escogido LoRa y el protocolo LoRaWAN para el desarrollo de nuestro dispositivo. Además, en el desarrollo del sistema trabajamos con las plataformas The Things Networks e IFTTT.

Abstract

This project consists of the creation of a compact, low-cost and highly energy-efficient device that will use IoT technologies. This IoT device will consist of sending an alert to the cloud which, after processing it and obtaining our location, will perform an action previously configured by the user himself. This alert is activated by a sequence introduced by the user using a button. This system will allow a wide range of actions from sending the location of the device by Telegram to activating the security alarm of a house if the device is inside it. An example of use would be:

- A user configures the system that as soon as the sequence is entered, the location of the device must be sent by SMS to his partner's phone.

-
- When the time comes, the user enters the sequence in the device and the alert is activated.
 - This is received by the server, which processes it and obtains the location of the device.
 - And finally, the system sends by SMS the location of the device to the established phone.

Also, in the project, the best known and most expanded wireless technologies used in IoT are summarized. With this, an analysis is made to determine the reason why LoRa and the LoRaWAN protocol was chosen for the development of our device. In addition, we work with the platforms The Things Networks and IFTTT during the development of the system.

Palabras clave:

- Internet de las cosas
- Arduino
- Tecnologías IoT
- SigFox
- LoRa
- LoRaWAN
- The Things Network (TTN)
- IFTTT
- Heltec WiFi Kit LoRa 32 v2
- Geolocalización

Keywords:

- Internet of things
- Arduino
- IoT technologies
- SigFox
- LoRa
- LoRaWAN
- The Things Network (TTN)
- IFTTT
- Heltec WiFi Kit LoRa 32 v2
- Geolocation

Índice general

1	Introducción	1
1.1	Motivación	1
1.1.1	Propuesta	1
1.2	Metodología y fases del proyecto	2
1.3	Objetivos	3
1.4	Organización del proyecto	4
2	Fundamentos teóricos y conceptos previos	7
2.1	Conceptos Previos	7
2.1.1	IoT	7
2.1.2	Arduino	10
2.1.3	Plataformas de gestión de eventos	15
2.2	Fundamentos Tecnológicos	18
2.2.1	Corto Alcance	19
2.2.2	Medio Alcance	22
2.2.3	Largo Alcance	23
2.3	Estudio de viabilidad	25
2.3.1	Elección de tecnología	26
3	Estudio de antecedentes o de alternativas	29
3.1	Digital Matter Guppy LoRaWAN	29
3.2	MOKOSmart	30
3.2.1	LW004-PB	31
3.2.2	LW004-CT	32
3.3	AlertCops	33
4	Análisis y diseño	35
4.1	Análisis y diseño del proyecto	35

4.1.1	Requisitos	35
4.1.2	Análisis	36
4.1.3	Diseño del proyecto	39
5	Implementación e integración	47
5.1	Aplicativo offline	47
5.1.1	Hardware	47
5.1.2	Programa	48
5.1.3	Interfaz gráfica	54
5.2	Aplicativo interno	55
5.3	Aplicativo online	60
6	Pruebas	63
6.1	Pruebas funcionamiento general	63
6.2	Pruebas obtención localización	66
7	Planificación y evaluación de costes	69
7.1	Planificación	69
7.1.1	Incidencias	70
7.2	Evaluación de costes	72
8	Conclusiones	75
8.1	Conclusiones	75
8.2	Líneas Futuras	77
A	Material adicional	81
A.1	Creación de una receta en IFTTT	81
	Lista de acrónimos	87
	Glosario	89
	Bibliografía	91

Índice de figuras

1.1	Representación de un sprint.	4
2.1	Red IoT	8
2.2	Gama Cortex-M ARM	9
2.3	Placa Arduino Uno	10
2.4	Shield vacía	12
2.5	IDE Arduino	14
2.6	”Hello, world!” Arduino	15
2.7	Topologías.	19
3.1	Guppy LoRaWAN	29
3.2	MOKOSmart LW004	31
3.3	Captura de la aplicación AlertCops	34
4.1	CU01 - Definición de la secuencia.	37
4.2	CU02 - Introducción de la secuencia.	38
4.3	Módulo WiFi LoRa 32 KIT v2. Parte frontal.	40
4.4	Módulo WiFi LoRa 32 KIT v2. Parte trasera.	40
4.5	Módulo NodeMcu V3 Lua WiFi.	41
4.6	Módulo RFM95W.	42
4.7	Arquitectura típica de una red LoRaWAN.	43
4.8	Consola TTN.	44
4.9	Esquema básico del sistema.	45
5.1	Diagrama de pines de WiFi LoRa 32 Kit v2.	48
5.2	Circuito esquemático.	48
5.3	Circuito.	49
5.4	Gestor de tarjetas.	50

5.5	Librería Heltec ESP32.	50
5.7	Diagrama de estado del algoritmo.	54
5.8	Interfaz prototipo principal.	55
5.9	Interfaz prototipo principal 2.	55
5.10	Interfaz prototipo Pop up.	55
5.11	TTN Crear aplicación.	56
5.12	TTN Registrar dispositivo.	57
5.13	TTN Dispositivo.	58
5.14	TTN Data Storage.	58
5.15	TTN Collos.	59
5.16	TTN LoRa Cloud.	60
5.17	IFTTT Configuración acción.	62
6.1	Captura 1 de las pruebas generales.	64
6.2	Captura 2 de las pruebas generales.	64
6.3	Captura 3 de las pruebas generales.	65
6.4	Captura 4 de las pruebas generales.	65
6.5	Mapeado del muestreo.	67
7.1	Diagrama de Gantt.	69
7.2	Mapa de la cobertura de la ciudad de La Coruña y su periferia.	72
A.1	IFTTT Paso 1.	81
A.2	IFTTT Paso 2.	82
A.3	IFTTT Paso 3.	82
A.4	IFTTT Paso 4.	83
A.5	IFTTT Paso 5.	84
A.6	IFTTT Paso 6.	85

Índice de tablas

2.1	Tabla diferencias Corto Alcance.	22
2.2	Tabla diferencias Medio Alcance.	23
2.3	Tabla diferencias Largo Alcance.	25
2.4	Tabla diferencias SigFox vs LoRa/LoRaWAN.	27
3.1	Tabla diferencias Guppy.	30
3.2	Tabla diferencias LW004-PB.	32
3.3	Tabla diferencias LW004-CT.	33
6.1	Tabla muestreo localizaciones.	66
7.1	Tabla coste recursos humanos.	72
7.2	Tabla coste recursos materiales.	73
7.3	Tabla coste final.	73

Introducción

EN este capítulo hablaremos sobre el problema objetivo de la sociedad actual. Además, abordaremos los objetivos para buscar una solución a dicho problema. Por último, enunciaremos la estructura y la organización del proyecto.

1.1 Motivación

En la actualidad el incremento de delitos tales como: robos, agresiones o secuestros se ha convertido en una lacra para la sociedad moderna que ve convertido su idílico futuro de paz y tranquilidad en miedo, inquietud e incluso venganza. La inseguridad surgida a partir de estos actos puede, incluso llegar a provocar grandes movimientos migratorios de personas en busca de ese lugar seguro y sosegado. Este incremento a menudo viene acompañado de una economía sumergida que lo apoya con un fin lucrativo como recompensa. Dicha economía y sus acciones son perseguidas por las fuerzas de seguridad, tanto públicas como privadas, para ponerles fin. El principal inconveniente al que se ven expuestas dichas fuerzas de seguridad es la saturación y la lentitud de la justicia para dictaminar sentencias. Este inconveniente provoca, en la mayoría de los casos, un cúmulo de retrasos que pueden alcanzar varias décadas para conseguir una resolución. A partir de este problema surgen empresas e iniciativas para buscar una solución viable o dar apoyo a las fuerzas de seguridad. Algunas de esas propuestas se mencionarán y explicarán brevemente en apartados posteriores.

1.1.1 Propuesta

Para el problema citado en el apartado anterior, este proyecto presentará una solución viable y práctica que prevé, desde dar soporte a las fuerzas de seguridad hasta llegar a facilitar una salida pacífica a una situación conflictiva o criminal. La solución tendrá como eje central un dispositivo hardware de pequeño tamaño, véase las dimensiones tales como un reloj de bolsillo, el cual tendrá un pulsador acoplado. Este dispositivo estará conectado a una

aplicación web a través de Internet. Dicha aplicación tendrá toda la información relacionada con el dispositivo, desde el porcentaje de batería hasta la ubicación del mismo. Con esta información junto con la proporcionada por el usuario que incluirá una secuencia de pulsado, el conjunto realizará una acción predefinida por el usuario tal como proporcionar la ubicación en ese momento de la pulsación a las fuerzas de seguridad o enviar un mensaje con la ubicación a un contacto de emergencia. Para que esta solución abarque el mayor rango de personas se ha buscado unas tecnologías que tengan un bajo coste de adquisición, un consumo energético reducido permitiendo tener unas baterías con duración de al menos varios días y una conectividad globalizada. Por ello esta solución estará basada en tecnologías, protocolos y hardware que se encapsulan dentro del concepto de Internet of Things (IoT). Sobre este concepto se hablará y explicará con más detalle en el apartado de fundamentos teóricos y concepto previos.

1.2 Metodología y fases del proyecto

En nuestro proyecto utilizaremos la metodología Scrum [1]. Esta es una metodología de desarrollo ágil tanto hardware como software. Consiste en un desarrollo incremental basado en la creación y asignación de objetivos. Estos objetivos se llevan a cabo por fases incrementales independientes permitiendo así un trabajo paralelo e incluso solapamiento entre estas fases. La metodología Scrum se centra en la consecución de varios puntos:

- Product Backlog: es la lista de funcionalidades que debe tener el producto ordenadas de mayor a menor relevancia.
- Sprint Backlog: es un índice de los puntos del product backlog que se deben cumplir en este sprint (iteración).
- Daily Scrum: son las reuniones diarias de pequeña duración que se realizan para organizar el trabajo que se va a realizar ese mismo día.
- Sprint Review: es la revisión que se lleva a cabo tras la finalización de un sprint sobre los avances del proyecto.
- Sprint Retrospective: Al finalizar el proyecto se realiza una revisión de los objetivos cumplidos recapitulando los errores encontrados durante el desarrollo para anotarlos y así evitarlos en futuros proyectos.

Además de estos puntos y para asegurar un correcto funcionamiento de Scrum, a los miembros del proyecto se le asignan roles. Estos son:

- Product Owner: es el administrador y gestor del Product Backlog y el responsable principal del proyecto.
- ScrumMaster: es la persona que se encarga de asegurar el correcto funcionamiento Scrum y que se cumpla con los objetivos del sprint.
- Equipo de desarrollo: son las personas encargadas de llevar a cabo las tareas descritas en el Sprint Backlog.

Scrum fue escogido a este proyecto por la necesidad de desarrollarlo por fases llegando a ser algunas de ellas igual salvo alguna mejora en un punto específico del Product BackLog. Por ejemplo, en una fase se implementará un pequeño programa de prueba en el dispositivo para familiarizarnos con el entorno y en una fase posterior se repetirá el programa pero añadiendo algún componente y sus cambios derivados.

Siguiendo esta metodología en nuestro proyecto asignamos todos los roles a una misma persona debido a que el proyecto sólo será desarrollado por una persona. Tanto el Product BackLog como el Sprint BackLog estarán plasmado en una pizarra magnética física en la zona de desarrollo. Además y como consecuencia de que el proyecto sólo cuente con un miembro, todas las fases serán desarrolladas de manera secuencial.

Una representación de los apartados de un sprint sería la que se ve en la imagen 1.1.

1.3 Objetivos

El principal objetivo de este proyecto es dar una solución viable de bajo coste con alta eficiencia energética. Es decir, queremos obtener un dispositivo barato, compacto y que consuma muy poco. Este dispositivo tiene que permitirnos enviar una alerta a la nube para procesarla y con ello, realizar una acción configurada previamente por el usuario. Todo esto, sin tener que preocuparnos de la batería, al menos, en varios años.

Para ello, se irán completando diferentes hitos a lo largo del proyecto. Estos hitos son los siguientes :

- Realización de un estudio sobre las tecnologías y protocolos IoT identificando los pros y los contras de cada uno de ellos para su posterior comparativa.
- Análisis del estudio y de la comparativa anterior para escoger la tecnología que mejor se adapte a nuestra solución.
- Diseño de la arquitectura de la solución a partir de los resultados.
- Implementación e integración del sistema diseñado previamente.
- Validación y pruebas del sistema piloto.

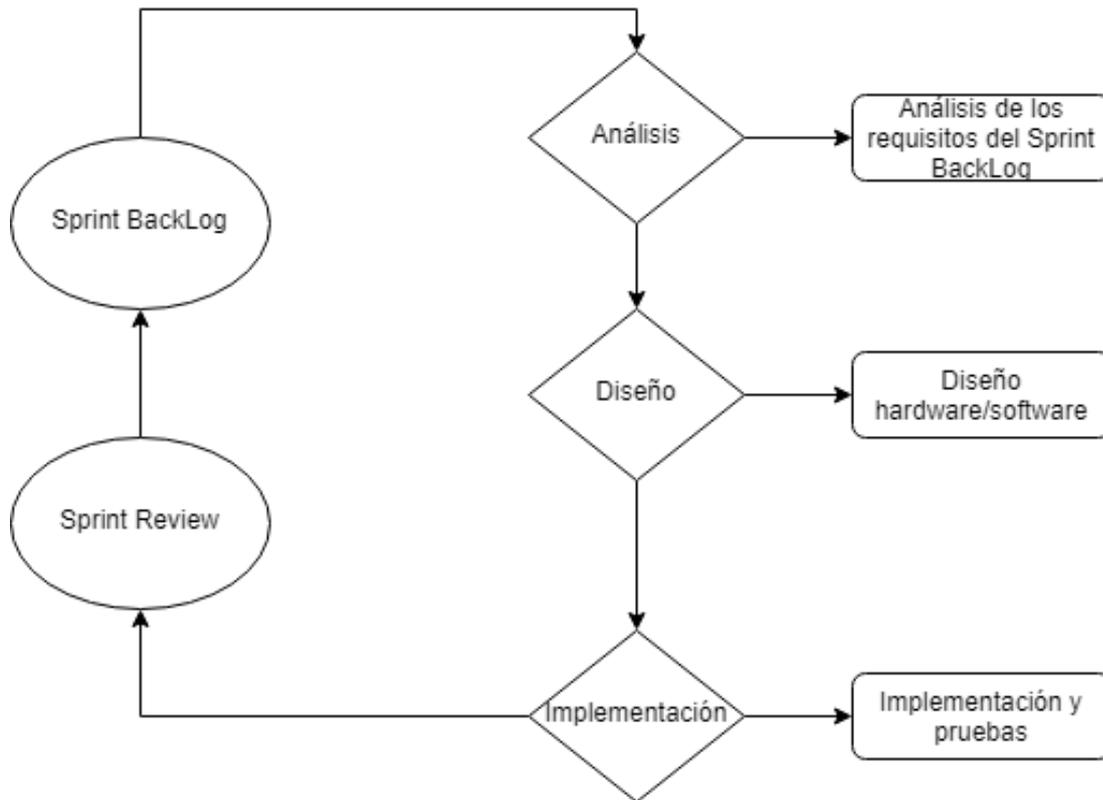


Figura 1.1: Representación de un sprint.

1.4 Organización del proyecto

Esta memoria consta de 8 capítulos en los que se profundizará en el marco teórico del proyecto, el diseño, su desarrollo y su posterior implementación. Ésta, comienza con la introducción, capítulo en el que nos encontramos actualmente, y continuará con:

- Fundamentos teóricos y conceptos previos: En este capítulo ubicaremos nuestro proyecto dentro de un marco teórico científico y explicaremos los conceptos relacionados con el mismo. Además, contará con un estudio de las tecnologías inalámbricas más importantes y con un análisis de viabilidad.
- Análisis de antecedentes, o alternativas: Este capítulo estudiará todas las soluciones existentes o alternativas a nuestro proyecto y realizará una comparativa entre ellas.
- Análisis y diseño: Este capítulo abarcará el análisis de los requisitos y el diseño de la solución propuesta por nuestro proyecto.
- Implementación e integración: En este capítulo se desarrollará el proyecto explicado paso a paso junto con su ensamblaje hardware-software.

- Pruebas: Este capítulo mostrará la puesta en funcionamiento del sistema en su conjunto y las posteriores pruebas que se le realizarán para verificar su correcto funcionamiento.
- Planificación y evaluación de costes: En este capítulo explicaremos la planificación que hemos seguido durante el proyecto junto con las demoras surgidas en ella. A mayores, realizaremos el cálculo del coste estimado y del coste real del proyecto.
- Conclusiones: Último capítulo en el que se expondrán las conclusiones surgidas tras la realización de este proyecto junto con el trabajo a seguir para mejorar, ampliar u ofertar el producto resultante.

Fundamentos teóricos y conceptos previos

EN este capítulo abordaremos los conceptos básicos y necesarios para poder comprender el marco teórico en el cual se ubicará el proyecto. Hablaremos sobre el concepto Internet of Things (IoT) y Arduino. Además, estudiaremos las principales tecnologías y protocolos dentro del concepto IoT para posteriormente realizar una comparativa y explicar las razones de porque hemos escogido LoRa frente al resto de alternativas.

2.1 Conceptos Previos

2.1.1 IoT

Para poder comprender el concepto de Internet of Things (IoT) [2], debemos tener en cuenta que no nos referimos a una sola tecnología o una única solución sino a un conjunto de dispositivos y objetos interconectados entre si a través de una red. Esta interconexión se realiza gracias a numerosos factores, como las tecnologías o los dispositivos que interactúan entre sí para crear una gran red con una única finalidad: conectar el máximo número de objetos entre sí sin la necesidad de la intervención humana 2.1. Esto significa que su objetivo es la interacción machine to machine (M2M). Esta interacción machine to machine consiste en realizar un intercambio de información, se comuniquen y se envíen datos, entre dos dispositivos electrónicos de manera autónoma, es decir, sin que haya ningún tipo de interacción humana durante el proceso de intercambio de información.

Cabe destacar que este concepto es el nivel base de la comunicación de Internet of Things. IoT se forma a través de la integración de múltiples dispositivos M2M junto con aplicativos y plataformas webs para el procesamiento de la información recopilada.

Internet of Things se asenta sobre unos principios básicos fundamentales que le permiten

2.1.1.1 Eficiencia energética

Todas la tecnologías empleadas en IoT tienen en común su reducido tamaño, dando la posibilidad de crear dispositivos equivalentes al tamaño de una moneda, y un consumo energético tan pequeño que permite tener baterías con una vida útil media de 10 años.

Relacionado con esto, surge durante el primer trimestre del año 2016 un nuevo módulo de memoria RRAM [3] llamado Moneta [4]. Este módulo de memoria no volátil (NVM) consume una quinceava parte de energía de su homólogo más cercano, aportando así un aumento en el ciclo de vida útil de una batería de hasta 10 años. Esta aparición revolucionó el apartado de autonomía en el ámbito IoT debido a que establecería, teóricamente, el nuevo límite de vida útil de una batería alrededor de los 25 años.

Con el fin de mejorar la eficiencia energética también aparecen nuevos procesadores diseñados especialmente para desarrollo de soluciones IoT que reducen su tamaño con respecto a sus predecesores y consiguiendo un ínfimo consumo de energía. Un claro ejemplo de estos procesadores son los procesadores RISC de 32 bits de ARM, los Cortex-M 2.2.

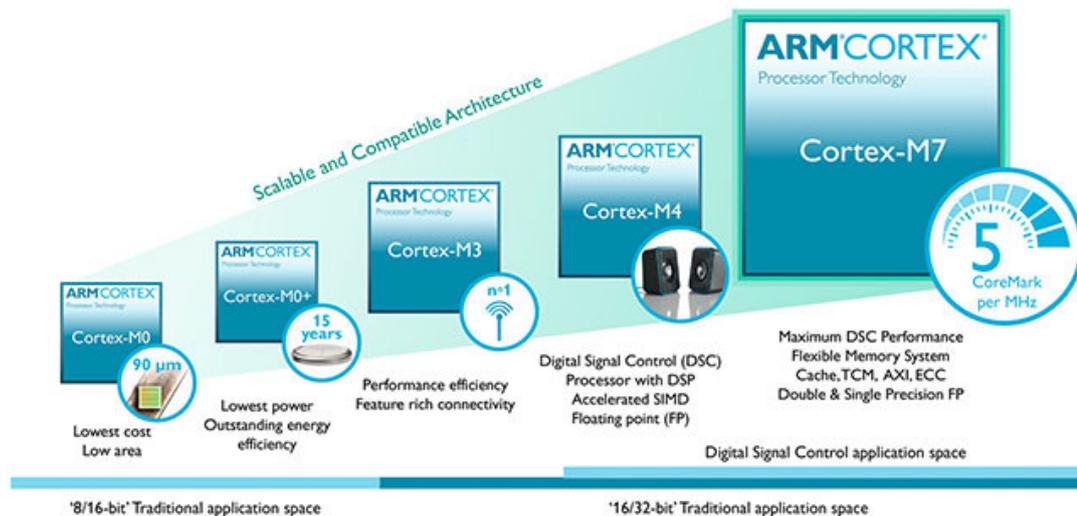


Figura 2.2: Gama Cortex-M ARM

Estas mejoras y progresos en el ámbito energético habilitan la posibilidad, tanto a empresas como a usuarios finales, de reducir gastos de mantenimiento en sus soluciones o dispositivos IoT y de necesitar un menor aprovisionamiento de baterías para su funcionamiento. Este abaratamiento y dicha posibilidad de tener un dispositivo completamente funcional y autónomo con una vida útil que ronda los 10 años es el mayor atractivo para el gran concepto de Internet of Things (IoT).

2.1.2 Arduino

Arduino [5] comenzó inicialmente como un proyecto de investigación Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino y David Mellis a principios de la década del 2000. Este consistía en proporcionar una forma fácil y económica para que principiantes y profesionales en el ámbito del hardware crearan dispositivos que pudieran interactuar con su entorno mediante sensores y actuadores. El proyecto se basaba en otros dos: el proyecto Processing[6] y el proyecto Wiring [7]. El proyecto Processing lo usaron como base para su entorno de desarrollo y, a su vez, utilizaron la estructura del lenguaje de programación del proyecto Wiring para su lenguaje. Posteriormente expondrían públicamente el proyecto, convirtiéndolo así en el primer proyecto de hardware de código abierto generalizado. Este proyecto permite construir dispositivos digitales e interactivos a través de unas placas de desarrollo de hardware (placas Arduino) 2.3. Estas placas están compuestas de unos componentes básicos para operaciones de E/S y un microcontrolador ensamblados en una placa PCB junto con pines que permiten la conexión de agentes externos que interactuarán con ella tales como sensores o actuadores.

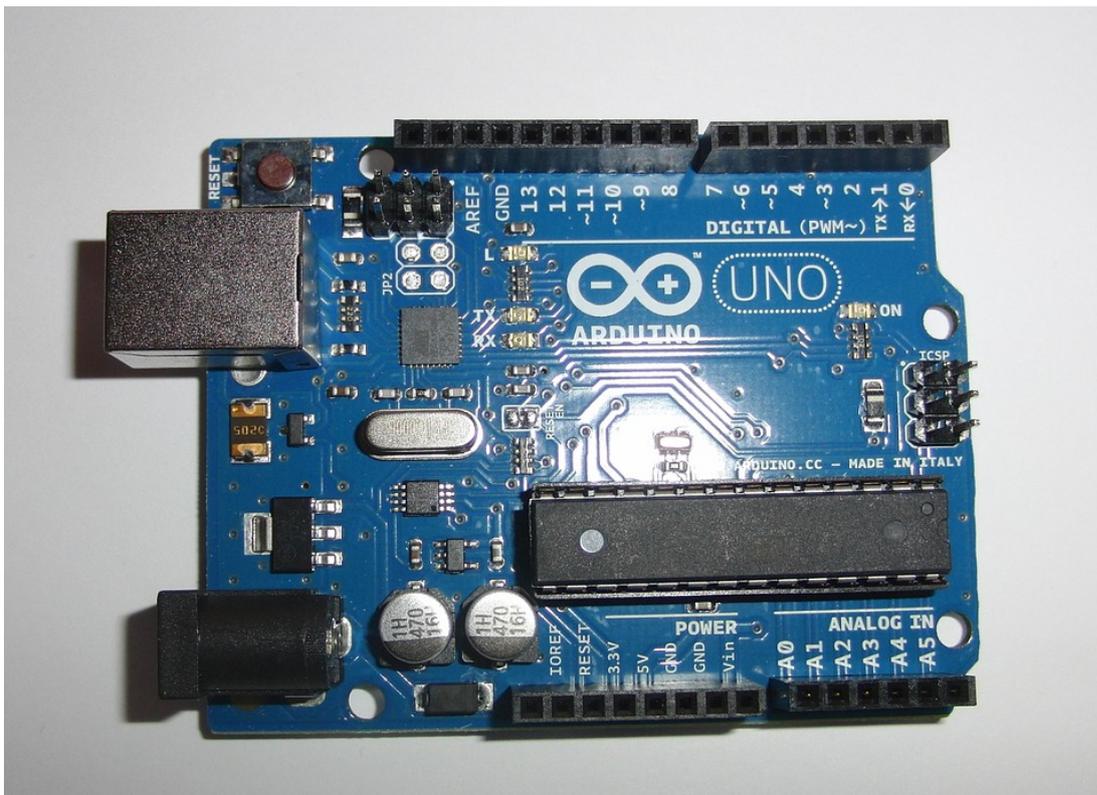


Figura 2.3: Placa Arduino Uno

Actualmente, el proyecto Arduino [5] es una fundación de código y hardware abierto

bajo las licencias GPL [8] y LGPL [9]. Esto permite que haya una gran variedad de placas de diferentes fabricantes teniendo una infinidad de opciones de compra e incluso teniendo la posibilidad de fabricar una tú mismo.

La enorme flexibilidad que aporta Arduino permite que puedas conseguir o fabricar placas para cualquier tipo de solución desde luces automáticas en una casa hasta un control de proximidad para un automóvil. Pero para ello, no sólo es necesario el hardware sino también el software necesario para llevar a cabo el programa de la solución deseada. Por eso, necesitamos conocer tanto el hardware como el saber desarrollar el software que se necesita para las placas Arduino.

2.1.2.1 Hardware

Las placas Arduino cuentan con unos componentes hardware comunes entre todas ellas y conforman el núcleo de toda placa. Estos son:

- El microprocesador: es el cerebro de la placa y esencial en el funcionamiento de la misma ya que se encarga de ejecutar el programa.
- La memoria: es la encargada de guardar el programa de la placa y la información durante la ejecución del mismo. En la placa se encontrarían dos tipos de memoria: la volátil tipo SRAM y la no volátil tipo EEPROM o MEMORIA FLASH.
- Los pines: son los elementos para transmisión y/o recepción de datos y los hay de dos tipos: analógicos y digitales.

Además de la placa principal, existen otros tipos de componentes llamados módulos o shields 2.4 que se trata de unas placas que se conectan a la principal para añadirle funcionalidades extra como puede ser GPS, WiFi, radio o pantalla LCD.

2.1.2.2 Software

En cuanto al software, hay que hablar del lenguaje de programación en el que se desarrollará y el entorno en el que se llevará a cabo.

2.1.2.2.1 Lenguaje La plataforma Arduino utiliza un lenguaje propio de medio nivel basado a partir del lenguaje de alto nivel Processing que es similar a C++ [10]. Al ser un lenguaje de medio nivel trabajo tanto con objetos básicos como caracteres o números como con bits y direcciones de memoria.

Arduino también parte del lenguaje C [11], lo que le permite soportar todas las funciones del estándar C [12] e incluso algunas funciones del lenguaje C++ [10]. A mayores de las fun-

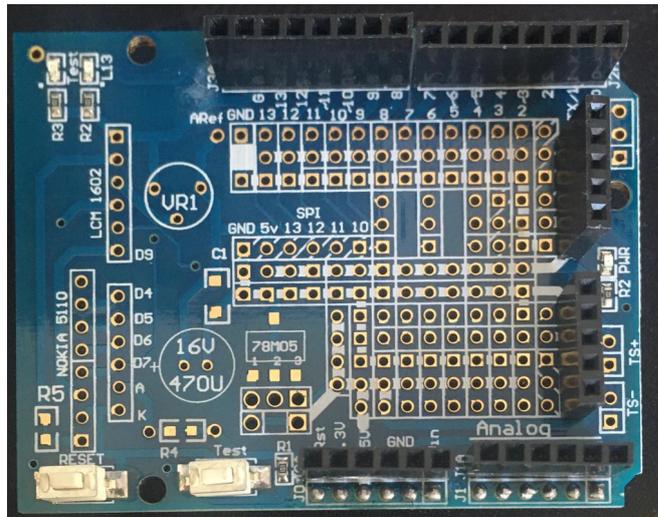


Figura 2.4: Shield vacía

ciones que soporta o comparte con otros lenguajes, el lenguaje propio de Arduino tiene sus funciones exclusivas para interactuar con la placa.

El lenguaje básico de Arduino [13] se puede dividir en 3 grandes partes:

- La estructura: Se basa en la estructura de C++.
 - Sketch - loop() y setup().
 - Estructuras de control como do...While, for o switchCase.
 - Operadores aritméticos como * (multiplicación).
 - Operadores de comparación como <= (menor o igual a).
 - Operadores booleanos, de bits y con punteros de acceso.
 - Operadores compuestos como ++(incremento).
- Los valores (variables y constantes).
 - Funciones de conversión de datos.
 - Calificadores de tipos de datos.
 - Tipos de datos incluyendo básicos como boolean o int hasta algunos más complejos como word o array.
- Las funciones: se utilizan para manejar la placa y llevar a cabo los cálculos necesarios en el programa. Hay de diferentes tipos y con diferentes finalidades como:
 - Controlar el tiempo como delay().

- Gestionar la E/S como `analogRead()` o `pinMode()`.
- Permitir la comunicación como `serial`.
- Manejar interrupciones como `attachInterrupt()`.
- Trabajar con bits y bytes como `bitSet()`.
- Matemáticas como `pow()` o `tan()`.

Aparte del lenguaje básico, el lenguaje consta de otro factor muy importante dentro de su comunidad, son las librerías [14]. Existen unas librerías estándar de Arduino que proporciona al programador junto con su Integrated Development Environment (IDE) y que le permiten interactuar con los elementos básicos y comunes en las placas. A estas, se le suman las múltiples librerías que en muchos casos, cada fabricante de un dispositivo electrónico, suele implementar para permitir sacar todo el rendimiento a su diseño y ahorrar quebraderos de cabeza al programador. Esta agregación permite la expansión de Arduino aportando nuevas funcionalidades o mejorando otras ya existentes que permiten interactuar mejor con uno u otro hardware que se añade a la placa. La plataforma Arduino categoriza oficialmente las librerías en:

- Comunicación.
- Procesamiento de datos.
- Almacenamiento de datos.
- Display.
- Sensores.
- Señal de E/S.
- Tiempo.
- Control del dispositivo.
- Otros.
- Sin categorizar.

En la actualidad, la plataforma Arduino reconoce y categoriza más de 3000 librerías de todo tipo de fabricantes o usuarios. Aparte de estas librerías, Arduino permite utilizar librerías propias u otras no reconocidas siempre y cuando sean compatibles.

2.1.2.2.2 Entorno de desarrollo La plataforma Arduino cuenta con su propio Integrated Development Environment (IDE) de manera abierta y oficial para todos los usuarios. Esta IDE cuenta con su versión online y su versión de escritorio tanto para sistemas UNIX como MACOS o WINDOWS.

La versión de escritorio es la más utilizada por los usuarios. Tiene una interfaz muy simple e intuitiva, además, al abrir un nuevo proyecto nos muestra el esqueleto básico de la estructura de un programa de Arduino 2.5. Esta IDE viene acompañada de múltiples ejemplos comentados que ayudan a iniciarse en Arduino. También aporta un administrador de librerías que permite buscar y añadir librerías reconocidas por Arduino facilitando así la adhesión de hardware a nuestros proyectos.

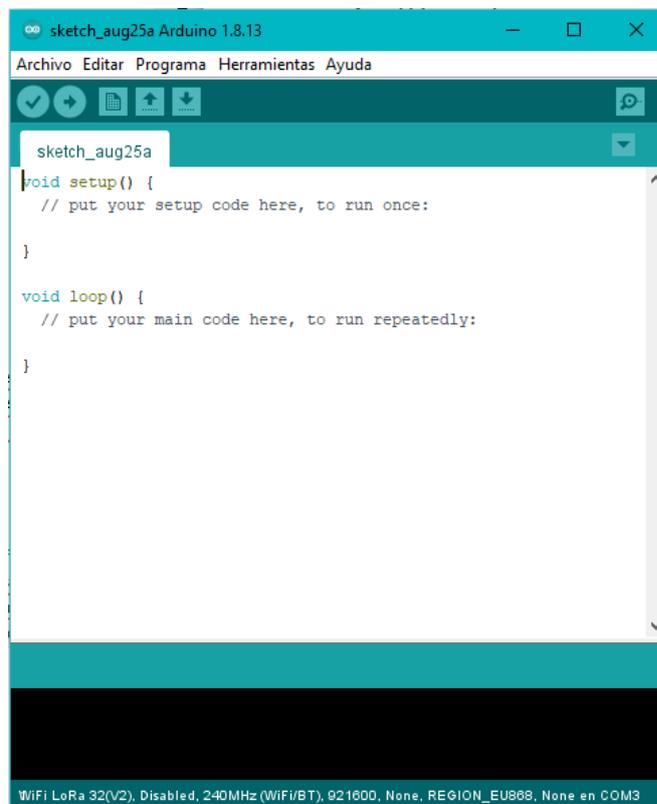


Figura 2.5: IDE Arduino

2.1.2.2.3 Ejemplo básico Por último y para ayudar a visualizar como sería un breve y simple programa en Arduino. Este ejemplo mostrará el texto "Hello world!" por una pantalla LCD. El código sería:

```
1 // include the library code:
2 #include <LiquidCrystal.h>
3
4 // initialize the library by associating any needed LCD interface
   pin
5 // with the arduino pin number it is connected to
6 const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
7 LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
8
9 void setup() {
10 // set up the LCD's number of columns and rows:
11 lcd.begin(16, 2);
12 // Print a message to the LCD.
13 lcd.print("hello, world!");
14 }
15
16 void loop() {
17 // set the cursor to column 0, line 1
18 // (note: line 1 is the second row, since counting begins with 0):
19 lcd.setCursor(0, 1);
20 // print the number of seconds since reset:
21 lcd.print(millis() / 1000);
22 }
```

El código utiliza en primera instancia la librería de la pantalla para facilitar la programación y la correspondiente asignación de los pines necesarios para la pantalla LCD. Con esto compilado y ejecutado se vería así:



Figura 2.6: "Hello, world!" Arduino

2.1.3 Plataformas de gestión de eventos

En la actualidad, existen múltiples aplicaciones y plataformas, en su mayoría gratuitas, que permiten centralizar, automatizar y conectar entre sí diferentes sistemas, dispositivos y

tecnologías. Su objetivo es facilitar al usuario la administración y automatización de sus dispositivos y tareas, favoreciendo la digitalización de la vida cotidiana. Este aspecto combinado con IoT, permite la creación de casas domóticas totalmente automáticas.

Las plataformas más usadas y conocidas en el entorno que nos interesa para nuestro proyecto, IoT son:

- OpenHab
- Home Assistant
- Node-Red
- IFTTT

2.1.3.1 OpenHab

OpenHab [15] es una plataforma de código abierto surgida en 2010. Fue desarrollada en Java basándose principalmente en el framework Eclipse SmartHome. Además, consta de Apache Karaf y Eclipse Equinox para crear un entorno OSGi que junto a su software altamente modular le permiten adquirir una gran cantidad de competencias a través de los add-ons. La plataforma tiene como objetivo permitir la integración de diferentes sistemas de automatización dentro de un mismo lugar proporcionando interfaces de usuario centralizadas, uniformes y con un enfoque común entorno a las reglas de automatización de todo el sistema. Su mayor ventaja es su independencia de los proveedores, de los fabricantes, del hardware y de los protocolos de comunicación que se intenten integrar. Su gran interoperabilidad proporciona al usuario un gran variedad de combinaciones de las diferentes soluciones existentes del mercado.

OpenHab divide su funcionamiento en dos grandes partes:

- Físico: es el apartado compuesto por los dispositivos que forman el sistema.
- Funcional: es el punto encargado representar la información de los dispositivos y sus conexiones en la interfaz del usuario.

2.1.3.2 Home Assistant

Home Assistant [16] es un software abierto de gestión de domótica desarrollado en Python que hizo su primera aparición a mediados del 2013. Este software se centra en la capacidad de integrar una gran cantidad de servicios y dispositivos de diferentes fabricantes en un entorno cerrado teniendo así privacidad y un control local total. Su versatilidad y debido a desarrollo en Python le proporciona una gran compatibilidad con diferentes dispositivos y sistemas operativos.

Este software proporciona al usuario unas acciones y scripts de sistemas basados en reglas que le permiten crear múltiples automatizaciones con:

- Control de tiempo
- Control de eventos/sucesos
- Notificaciones
- Control de voz
- Tareas de programación

2.1.3.3 Node-Red

Node-Red [17] es una plataforma de desarrollo de código abierto que se ejecuta en Node.JS, que a su vez es un motor de ejecución asíncrono de Javascript para servidores. Esta plataforma se basa en un entorno de desarrollo gráfico (herramienta de programación visual) para la creación de aplicaciones JavaScript y su uso abarca desde pequeñas soluciones de domótica hasta grandes aplicaciones de IoT en la nube.

Node-Red ha cuajado como un editor de flujo de datos donde se puede manejar nodos permitiéndolos conectarse entre sí con el fin de que se comuniquen entre ellos. Esto permite conectar dispositivos hardware, APIs y servicios web y de Internet de una manera rápida, potente y sencilla. Estas características es lo que lo han convertido en un framework perfecto para la gestión y transformación de datos en tiempo real en entornos como IoT, Industry 4.0 o marketing digital. Su principal ventaja es su facilidad de uso ya que no requiere de conocimientos previos de programación y apenas necesita recursos de cómputo para funcionar adecuándolo a integrarse en dispositivos de bajo coste.

2.1.3.4 IFTTT

IFTTT[18] (IF This, Then That) es una plataforma de código abierto lanzada a finales del año 2010 que interconecta múltiples servicios para crear y programar acciones usando "recetas". Estas "recetas" son una combinación de sucesos que se tienen que dar de una determinada manera para llevar a cabo dicha acción. Un ejemplo práctico de una receta sería:

- Un usuario de la plataforma crea una "receta" en la cual determina que si Beyoncé publica una foto en Instagram, la plataforma debe avisarle al whatsapp mediante un mensaje establecido por él.
- Dado el momento, Beyoncé publica una foto en Instagram.
- La plataforma recibe el aviso de esa subida.

- La plataforma comprueba la receta y las condiciones establecidas.
- Las condiciones son cumplidas, por tanto, la plataforma envía al whatsapp del usuario el mensaje establecido.

En capítulos posteriores, se mencionará nuevamente debido a que formará parte del proyecto.

2.2 Fundamentos Tecnológicos

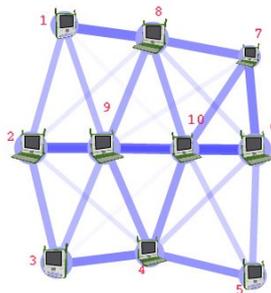
Dentro del concepto IoT existen múltiples tecnologías que se pueden utilizar para realizar una conexión entre dos dispositivos. Estas se podrían categorizar en tres grandes grupos que se establecen conforme al límite de alcance de la señal que ofrece cada una. Estas categorías son:

- Corto Alcance: < 200 metros.
- Medio Alcance: ≥ 200 metros y < 5 kilómetro.
- Largo Alcance: ≥ 5 kilómetro.

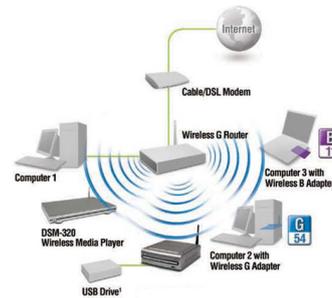
Antes de continuar explicando cada una de las categorías hay que conocer unos términos básicos para poder entender mejor el análisis y la comparativa. Estos términos son:

- Frecuencia: Es la frecuencia en la que trabaja la señal inalámbrica de esa tecnología en concreto. Se mide en hercios (Hz). Una particularidad es que cuando la frecuencia es inferior a 1GHz, en Europa (EU) se debe usar la frecuencia de 868MHz mientras que en América (US) se debe usar la frecuencia de 915MHz.
- Rango: Es el alcance que tiene la señal emitida por esa tecnología. Suele darse mediante un intervalo y con los valores medidos en entornos no urbanos.
- Precio del chipset: Es el dinero que tiene que desembolsar un usuario para conseguir un módulo o chip que permita trabajar con esa tecnología. Pueden ser de coste:
 - Bajo: $< 5€$.
 - Medio: $\geq 5€$ y $\leq 15€$.
 - Alto: $> 15€$.
- Topología: es la forma en la que se encuentra diseñada la red de esa tecnología. Es decir, como están conectados los nodos o dispositivos entre si. Hay varios tipos pero los que veremos son:

- Malla: Es una red en la cual cada nodo está conectado a todos los nodos. Esto implica mayor robustez frente al fallo de algún elemento de la red. 2.7a
- Estrella: Es el tipo de red en el que los nodos están conectados a un punto central y todas las comunicaciones deben pasar estrictamente necesario por ese punto. Esto crea una gran robustez frente a los fallos en los nodos pero crea un single point of failure (SPOF), es decir, que si es punto central falla, toda la red dejaría de funcionar. 2.7b



(a) Topología Malla.



(b) Topología Estrella.

Figura 2.7: Topologías.

A continuación, hablaremos de cada categoría y de las principales tecnologías que podemos encontrar en cada una.

2.2.1 Corto Alcance

En esta categoría se encuentran las tecnologías cuyo alcance no supera los 200 metros.

2.2.1.1 ZigBee

ZigBee es una tecnología abierta que es controlada por la ZigBee Alliance [19]. Trabaja con una red de malla en la frecuencia de 2.4 GHz (aunque puede funcionar en la frecuencia 915MHz (US) y la frecuencia 868MHz (EU)). En esta frecuencia consigue una velocidad de transmisión de datos de 250 kbps y un rango aproximado de 10 - 20 metros. Para que la red de esta tecnología funcione se necesita de 3 tipos de dispositivos:

- El coordinador ZigBee: es el elemento más importante de la red y el nodo más completo ya que se encarga de controlar la red y todos los caminos existentes en ella para su comunicación.
- El router ZigBee: es el dispositivo que interconecta los nodos.

- El dispositivo final ZigBee: es el nodo final que se encarga de recibir información y comunicarse con su nodo padre. Este dispositivo puede permanecer gran parte del tiempo en modo reposo y sólo interactuar durante determinados momentos lo que le permite alargar la duración de su batería.

Estos dispositivos a su vez se podría agrupar en dos tipos de nodos diferentes:

- Activos: Son los nodos que tienen una funcionalidad completa y permanecen en todo momento funcionando. En este grupo entrarían los routers y los coordinadores.
- Pasivos: Son los nodos con funcionalidad reducida y suelen ser sensores o actuadores de la red. En este grupo entrarían los dispositivos finales.

El precio de un chipset ZigBee sería de un rango medio, casi bajo, alcanzando en la actualidad un precio medio de 6€. Esta tecnología tiene como gran desventaja que necesita de demasiados dispositivos para asegurar un correcto funcionamiento de su red lo que implica una latencia relativamente alta.

2.2.1.2 Z-Wave

Z-Wave [20] es una tecnología cerrada propiedad de Silicon Labs [21]. Una ventaja de que esta tecnología sea cerrada es que todos los dispositivos compatibles con Z-Wave funcionan perfectamente entre ellos. Aunque tiene la desventaja que conseguir un dispositivo compatible supone un desembolso alto debido a sus precios.

Esta tecnología funciona con una red de malla en la frecuencia 915MHz (US) y en la frecuencia 868MHz (EU). Llega a alcanzar una velocidad de transmisión de 40 kbps y 20 kbps respectivamente. El alcance que puede llegar a dar Z-Wave es de 100 metros. Su funcionamiento es parecido a ZigBee y necesita de 3 tipos de dispositivos:

- Los controladores maestro: son los elementos que se encargan de la seguridad de la red y de controlar el enrutamiento.
- Nodos intermedios: es una unidad Z-Wave que interconecta los nodos pero que a su vez también puede ejercer de nodo final. Para poder llevar a cabo ambas funciones, este nodo debe tener suministro constante de energía como los controladores lo que le permitiría ejercer como enrutador.
- Nodos finales: es el nodo final que se encarga de recibir información y comunicarse con su nodo padre. Este dispositivo puede permanecer gran parte del tiempo en modo reposo y sólo interactuar durante determinados momentos lo que le permite alargar la duración de su batería.

2.2.1.3 BLE

Bluetooth Low Energy (BLE) es una tecnología de red de área personal inalámbrica diseñada y suministrada por Bluetooth Special Interest Group [22]. Esta es una asociación privada sin ánimo de lucro que busca promover el uso de Bluetooth. Por ello, si alguna compañía desea incorporar tecnología Bluetooth a sus productos o diseños debe formar parte de la asociación. Actualmente consta con más de 30000 miembros.

Volviendo a BLE, esta tecnología es una variante del Bluetooth tradicional. Esta variante reduce considerablemente el consumo energético necesario para trabajar manteniendo un alcance muy similar. Tanto BLE como Bluetooth tradicional operan en la banda de frecuencia de 2.4GHz pero BLE se mantiene en modo reposo todo el tiempo hasta que se inicia una conexión. Una diferencia destacada son los tiempos reales de conexión, mientras en BLE son unos pocos milisegundos, en el tradicional son más de 100 milisegundos. Otra diferencia a mencionar entre ambos es la velocidad de transmisión que en BLE se ve reducida frente a la tradicional, 30 kbps frente a 25 mbps. En cuanto al coste en BLE, cabe decir que es un punto a su favor ya que se encuentra en el rango de precios bajo siendo posible adquirir un chipset por menos de 5€.

2.2.1.4 Wi-Fi

Wi-Fi es la tecnología más conocida y extendida a nivel mundial. Pertenece a la Wi-Fi Alliance [23]. Surgió en abril del 2000 bajo la certificación de interoperabilidad de equipos según la norma IEEE 802.11b de 1999 [24]. Esta certificación aseguraba que todos los dispositivos, fuera cual fuera su fabricante, con el sello Wi-Fi podrían trabajar juntos sin ocasionar ningún tipo de problemas. Esta tecnología es la que mayor velocidad de transferencia de datos ofrece, pudiendo llegar a los 10 Gbps en la versión Wi-Fi 6 que sigue el estándar 802.11ax y que opera en las frecuencias de 2.4 GHz y 5 GHz. Esta tecnología funciona con una topología de estrella en la cual podemos encontrar dos tipos de dispositivos:

- Dispositivos de red: son los encargados de asegurar un correcto funcionamiento de la red. En este grupo se pueden diferenciar a su vez entre 3 subtipos:
 - Router: son dispositivos compuestos formados por un enrutador que interconecta redes diferentes, un conmutador y un punto de acceso. Su función principal es dar acceso a internet a todos los dispositivos que se conecten a él.
 - Repetidor: son dispositivos que se encargan de extender la cobertura de la red inalámbrica.
 - Punto de acceso: son los dispositivos necesarios para crear "esa red wifi" que permiten que otros dispositivos se conecten a ellos de manera inalámbrica.

- Dispositivos terminales: son los nodos finales de la red. Suelen ser tarjetas receptores que conectan móviles, ordenadores y otros dispositivos electrónicos.

La gran ventaja de esta tecnología es fama y sus años de experiencia que le aportan una cuota del mercado mayoritaria y con un coste de adquisición medio. A pesar de ello, esa sobremasificación de la tecnología Wi-Fi también es su desventaja debido a que en muchos lugares la banda de frecuencia en la que trabaja está tan saturada que se vuelve imposible lograr un correcto funcionamiento de la misma.

2.2.1.5 Comparativa

Ahora plasmaremos en una tabla los datos más relevantes de las tecnologías descritas con anterioridad para ver mejor sus diferencias 2.1.

Tecnología	Frecuencia	Velocidad máx	Rango	Coste	Abierto
ZigBee	2.4GHz, 915/868MHZ	250 kbps	20 metros	medio	SI
Z-Wave	915/868MHZ	40 kbps	100 metros	alto	NO
BLE	2.4GHz	30 kbps	60 metros	bajo	SI
Wi-Fi	2.4/5GHz	10 Gbps	80 metros	medio	SI

Tabla 2.1: Tabla diferencias Corto Alcance.

2.2.2 Medio Alcance

En esta categoría se encuentran las tecnologías cuyo alcance supera los 200 metros pero no sobrepasan los 5 kilómetros.

2.2.2.1 Weightless

Weightless [25] es un estándar abierto poco conocido que trabaja en el espectro sin licencia y que utiliza una topología de estrella. Hay 3 subtipos de esta tecnología:

- Weightless-W: en esta variante se usan las frecuencias sin licencias que se encuentran entre las bandas de las estaciones de TV.
- Weightless-N: utiliza las frecuencias 868(ES)/915(US)MHz.
- Weightless-P: emplea el rango de banda estrecha de 12.5 kHz con la posibilidad de comunicaciones bidireccionales.

Esta tecnología con las condiciones favorables puede llegar a tener un alcance de casi 4 kilómetros. En cuanto al coste de desarrollar un dispositivo en esta tecnología, hay que

mencionar que no es muy extendida por lo que hay poca variedad y ello implica un coste medio cercano a alto.

2.2.2.2 Wi-Fi HaLow

Wi-Fi HaLow es una variante de Wi-Fi presentado en el estándar IEEE 802.11ah del 2007. Esta variante mejora sustancialmente la calidad y cobertura de nuestra red Wi-Fi al trabajar en la banda de frecuencia de 900 MHz. Este cambio de frecuencia no sólo implica menos interferencias en la señal sino que también implica un menor consumo de energía por parte de los dispositivos lo que conlleva un aumento en el ciclo de vida de la batería.

Cabe destacar que no todo son mejoras en la variante. Como precio por reducir el gasto energético y bajar la banda de frecuencia, se reduce también su velocidad de transmisión de datos llegando a ser en las condiciones ideales de 347 Mbps.

Algo que no cambia con respecto a Wi-Fi es su funcionamiento. Sigue utilizando la topología en estrella y necesita de dos tipos de elementos en la red. Sin embargo, al ser relativamente reciente, todavía no se tiene una buena estimación del coste que supondrá esta tecnología, aunque la Wi-Fi Alliance anunció que sería similar al Wi-Fi.

2.2.2.3 Comparativa

A pesar de sólo haber mencionado 2 tecnologías (4 separando las variantes de Weightless) en esta categoría, realizaremos la comparativa entre ellas debido a que las consideramos suficientes en cuanto a medio alcance se refiere 2.2.

Tecnología	Frecuencia	Velocidad máx	Rango	Coste	Abierto
Weightless-P	12.5 kHz	200 bps	2 km	medio	SI
Weightless-N	915/868 MHz	100 kbps	4 km	medio	SI
Weightless-W	-	100 kbps	4 km	medio	SI
Wi-Fi HaLow	900 MHz	347 Mbps	4 km	medio	SI

Tabla 2.2: Tabla diferencias Medio Alcance.

2.2.3 Largo Alcance

En esta categoría se encuentran las tecnologías cuyo alcance supera los 5 kilómetros. Estas tecnologías se enmarcan dentro del término low-power wide-area network (LPWAN). Esto quiere decir que es un área de amplia envergadura de comunicación inalámbrica que necesita de una baja potencia para intercambiar información.

2.2.3.1 Narrowband IoT

Narrowband IoT o NB-IoT es una tecnología desarrollada por 3GPP [26] para cubrir el mercado de IoT y hacer frente a las otras LPWAN. NB-IoT funciona en la actual red móvil ya existente incluyendo la banda GSM, además de la misma banda que LTE y por tanto, compartiéndola con ella.

Es una tecnología abierta que nace condicionada por la actual infraestructura de LTE y la red móvil lo que le obliga a trabajar con una topología estrella celular sin introducir modificaciones. Su banda de frecuencias es inferior a 1 GHz, estando comprendida entre 700 MHz y 800 MHz normalmente. Esto le facilita la penetración en edificios llegando a una velocidad de transmisión de 100 kbps y tener un alcance en espacios abiertos de 20 km. A pesar de ello, su coste de adquisición el alto lo que ralentiza su expansión.

2.2.3.2 SigFox

SigFox [27] es una tecnología creada, desarrollada y distribuida por la empresa del mismo nombre. Esta es una de las 2 grandes tecnologías referentes en el campo de LPWAN junto con LoRa, un dúo que siempre es comparado debido a sus similitudes y su alta cobertura a nivel mundial.

Esta tecnología utiliza las bandas de frecuencia de 902MHz en América y la 868MHz en Europa consiguiendo una velocidad de transmisión de hasta 600 bps y un rango de alcance aproximado de 40 kilómetros. Al ser una tecnología cerrada por la empresa SigFox, el coste de adquisición y desarrollo viene marcado por ella siendo un precio alto. A pesar de ello, al ser cerrado te asegura la completa compatibilidad entre sus dispositivos y un correcto funcionamiento.

Con SigFox, un usuario sólo tendría que adquirir el dispositivo final para que la red funcione. La empresa aporta los dispositivos enrutadores, las puertas de enlace y demás dispositivos necesarios para que una red de topología en estrella funcione. Esto aporta simplicidad al usuario final y mayor comodidad de instalación al aportar un aplicativo web para la configuración del dispositivo.

2.2.3.3 LoRa

LoRa es una tecnología abierta de modulación de radio de espectro expandido patentada por Semtech [28] y administrada por LoRa Alliance [29] para LPWAN. La tecnología LoRa es una conexión punto a punto y funciona en las frecuencias de 868 MHz (EU), 915 MHz (US) y 433 MHz (Asia) consiguiendo una velocidad de transmisión de datos de hasta 50 kbps. El rango de LoRa en entorno abierto suele alcanzar los 20 o 30 kilómetros en circunstancias normales. Cabe destacar que si las condiciones lo permiten se pueden superar los 100 km con facilidad

llegando a alcanzar incluso los 210 km en tierra (record conexión tierra-tierra establecido en febrero de 2017) o alcanzando la friolera cantidad de 766 km de distancia utilizando un globo aerostático para establecer una conexión globo-tierra. Esta distancia es el actual record mundial de transmisión en LoRa siendo establecido el 13 de julio del 2019 [30].

Como LoRa es una tecnología punto a punto, LoRa Alliance propuso el protocolo de red LoRaWAN que administra y comunica dispositivos utilizando la tecnología LoRa. LoRaWAN a menudo se confunde con LoRa y viceversa pero hay que destacar que son dos términos diferentes pero complementarios entre si. Este protocolo utiliza una topología en estrella que se compone de dos tipos de dispositivos:

- Gateways: también llamados antenas son los encargados de recibir y enviar la información a los nodos.
- Nodos: son los dispositivos finales, tales como sensores y/o actuadores que intercambian información con el gateway.

Este protocolo aporta a LoRa las funcionalidades de una red compleja como SigFox pero siguiendo abierta lo que permite continuar con un coste de adquisición y desarrollo medio.

2.2.3.4 Comparativa

A pesar de existir más tecnologías en esta categoría, sólo realizaremos la comparativa entre las mencionadas ya que son consideradas las más importantes en cuanto a largo alcance se refiere 2.3.

Tecnología	Frecuencia	Velocidad máx	Rango	Coste	Abierto
NB-IoT	700-800 MHZ	100 kbps	20 km	medio	SI
SigFox	902/868 MHZ	600 bps	40 km	alto	NO
LoRa	915/868/433 MHZ	50 kbps	> 50 km	medio	SI

Tabla 2.3: Tabla diferencias Largo Alcance.

2.3 Estudio de viabilidad

Tras haber realizado el análisis e investigación de todas las tecnologías descritas en la sección anterior 2.2, ahora procedemos a escoger cual de ellas es la que mejor se adapta a nuestra propuesta de solución. Posteriormente, profundizaremos un poco más en la tecnología elegida.

2.3.1 Elección de tecnología

Para nuestra propuesta de solución necesitamos una tecnología que cumpla varias condiciones importantes. La primera es que la tecnología escogida debe poder abarcar grandes áreas de terreno manteniéndose dentro de la cobertura de la red. Teniendo en cuenta esta condición podemos descartar varios candidatos quedandonos sólo con 3 posibles candidatos: Narrow-band IoT (NB-IoT), SigFox y LoRa (con LoRaWAN). Estos candidatos son las tecnologías de largo alcance que habíamos seleccionado.

La segunda condición que la tecnología finalmente escogida debe cumplir es que debe ser asequible a todo el mundo y dar facilidades a la hora de desarrollar un producto. Teniendo en cuenta los 3 candidatos y observandolos detenidamente con esta condición nos damos cuenta que NB-IoT no la cumple debido a su alto coste, comunidad maker pequeña y una red compleja. Por tanto, nos quedaría escoger entre SigFox y LoRa(LoRaWAN).

Para finalizar, profundizaremos un poco más en ambas tecnologías para compararlas y facilitarnos una mejor toma de decisiones. En esta profundización añadiremos puntos como:

- El tamaño del paquete que admite la tecnología.
- La seguridad de la que dispone durante la comunicación.
- Si permite la localización.
- Si la comunicación es bidireccional.
- La sensibilidad máxima que tiene la tecnología.

Añadiendo los datos de ambas tecnologías a estos puntos obtenemos una tabla más completa que nos permite ver las diferencias entre SigFox y LoRa(LoRaWAN) 2.4.

Tras visualizar la tabla 2.4 y estudiarla al detalle nos encontramos con una tecnología claramente ganadora. Esto se debe a la principal condición que debía cumplir nuestra solución y es que debía ser lo más asequible posible y esto lo cumple la tecnología LoRa/LoRaWAN.

	SigFox	LoRa/LoRaWAN
Banda de frecuencia	868/902 MHz	868/915 MHz
Rango (urbano)	3-8 km	3-8 km
Rango (rural)	20-40 km	30-100 km
Velocidad	600 bps ¹	50 kbps ²
Tamaño paquetes	12 bytes	A definir por el usuario
Seguridad	NO	AES CCM (128-bit)
Payload	10 bytes	50 bytes
Mayor sensibilidad (dBm)	-142	-142
Bidireccional	SI	SI
Vida batería	10 años	> 10 años
Localización	SI	SI
Abierta	NO	SI
Subscripción	SI	NO
Coste	alto	medio
Tamaño comunidad maker	pequeña	muy grande

Tabla 2.4: Tabla diferencias SigFox vs LoRa/LoRaWAN.

Estudio de antecedentes o de alternativas

EN este capítulo analizaremos los proyectos existentes o que se encuentran en desarrollo que presentan una solución alternativa a la de este proyecto para dar solución al caso de uso planteado.

3.1 Digital Matter Guppy LoRaWAN

Digital Matter [31] es una empresa nacida en 2001 que desarrolla dispositivos GPS con IoT para múltiples usos desde la industria hasta el usuario final llegando a ser reconocidos y galardonados a nivel mundial. Entre todos sus productos, el que encajaría mejor en nuestro problema sería Guppy LoRaWAN [32]. Este dispositivo funciona como un localizador GPS de bajo consumo que utiliza LoRaWAN para comunicarse con el usuario o la plataforma destino. Este dispositivo de pequeño tamaño 3.1 está alimentado por dos baterías AAA alcalinas o de litio, reemplazables fácilmente por el usuario tras su ciclo de uso, que le otorgan una vida útil ininterrumpida teórica de poco más de 5 años por ciclo de baterías.



Figura 3.1: Guppy LoRaWAN

Guppy LoRaWAN consta de resistencia al agua y el polvo gracias a su carcasa plástica compacta. Además tiene incorporado un acelerómetro interno que permite detectar todo tipo de movimiento, utilidad que se puede usar para configurar una alerta en caso de movimiento dentro de unas determinadas horas.

Tras analizar a fondo el producto, podemos encontrar varios puntos donde nuestro proyecto mejora de cara al problema que queremos dar solución. Estos puntos son:

- La duración de la batería: A pesar de que el dispositivo tiene una vida teórica de 5 años por ciclo de batería nuestra solución tendría una teóricamente una vida útil superior a los 10 años por ciclo de batería.
- El tamaño: A pesar de que el tamaño de Guppy no es muy grande ya que ocupa poco más que dos pilas AAA, nuestra solución final (no el prototipo sin miniaturizar) equivaldría al tamaño de una moneda de dos euros.
- Iniciativa de alerta por parte del usuario: Este punto es algo que le falta a Guppy y es esencial.

Para facilitar la visión de las diferencias, lo plasmamos en la tabla 3.1.

Dispositivo	Batería	Tamaño	Iniciativa usuario
Guppy	5 años	dos pilas AAA	NO
Propio	10 años	moneda	SI

Tabla 3.1: Tabla diferencias Guppy.

3.2 MOKOSmart

MOKOSmart [33] es un fabricante profesional chino de dispositivos inteligentes que amplió sus fronteras y comenzó a investigar y desarrollar dichos dispositivos. Con este cambio pasó de ser un fabricante a ser una empresa innovadora y puntera que ofrece desde la fabricación del dispositivo electrónico hasta la programación de un firmware personalizado permitiendo así al cliente recibir un producto ya maduro que se podría vender directamente.

En la actualidad, MOKOSmart es un importante diseñador y proveedor de dispositivos inteligentes IoT además de ser miembro del LoRa Alliance [29]. LoRa Alliance es una asociación sin ánimo de lucro que busca implantar y promover el uso global de redes de largo alcance de baja potencia (LPWAN) IoT mediante la promoción y desarrollo del estándar abierto LoRaWAN. La asociación cuenta, actualmente, con más de 500 empresas miembros.

Entre su amplia gama de productos y soluciones IoT, destacamos dos dispositivos que se adecuarían al problema que queremos solucionar. LW004-PB y LW004-CT son casi idénticos externamente, salvo unos pocos milímetros diferentes en las dimensiones, pero internamente y en el firmware tienen importantes diferencias 3.2. Ambos dispositivos son de reciente creación surgiendo a mediados de mayo del 2020 y recibiendo algunas mejoras en junio del mismo año.

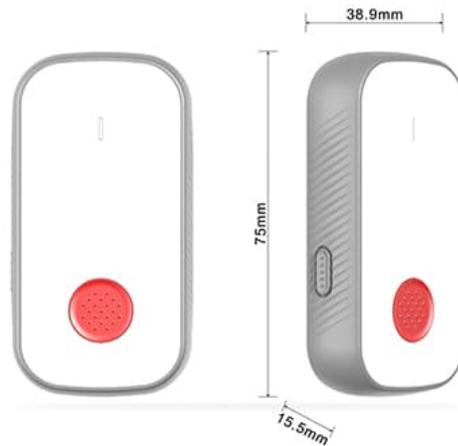


Figura 3.2: MOKOSmart LW004

3.2.1 LW004-PB

El dispositivo LW004-PB [34] fue diseñado específicamente para realizar un seguimiento tanto en interiores como exteriores. Consiste en un botón del pánico que tras su activación enviará la información de la ubicación al servidor mientras la alerta no se detenga. Utiliza el protocolo LoRaWAN para el intercambio de información con el servidor y para el cálculo de la ubicación utiliza GPS para los exteriores y BLE (Bluetooth Low Energy) para los interiores. También cuenta con un acelerómetro, un led para indicar la batería baja y un motor de vibración para funcionar durante la alerta.

Siguiendo la dinámica del análisis del anterior dispositivo y estudiar a fondo el dispositivo en cuestión, comparemos los mismos puntos entre el LW004-PB y nuestra solución siendo estos:

- La duración de la batería: Mientras que nuestra solución tendría una vida útil teórica superior a los 10 años por ciclo de batería, la del dispositivo LW004-PB no viene especificada y lo único que sabemos es que es una batería recargable.

- El tamaño: El tamaño es ligeramente inferior al dispositivo Guppy, pero nuestra solución propuesta seguiría siendo más pequeña.
- Iniciativa de alerta por parte del usuario: Ambas soluciones tienen este punto pero con la sutil diferencia de que nuestro botón puede hacer más de una función dependiendo de la secuencia introducida.

La tabla resultante de las diferencias es: 3.2.

Dispositivo	Batería	Tamaño	Iniciativa usuario
LW004-PB	-	inferior a Guppy	SI
Propio	10 años	moneda	SI

Tabla 3.2: Tabla diferencias LW004-PB.

3.2.2 LW004-CT

El dispositivo LW004-CT [35] fue diseñado para realizar un rastreo de contactos. Sus funciones principales sería las de rastrear un contacto cercano y la función de calcular la distancia de seguridad interpersonal avisando en caso de infracción de esta. El dispositivo utiliza el protocolo LoRaWAN para el intercambio de información con el servidor y para el cálculo de la distancia, rastreo y la ubicación en interiores utiliza BLE. También cuenta con un acelerómetro, un led para indicar la batería baja y un motor de vibración para funcionar durante la alerta.

Este dispositivo es bastante reciente y surgió como una solución y ayuda para mantener la distancia de seguridad durante la pandemia y la post-pandemia.

Siguiendo la misma dinámica y el posterior análisis del dispositivo, compararemos los mismos puntos deseados entre el LW004-CT y nuestra solución siendo estos:

- La duración de la batería: Mientras que nuestra solución tendría una vida útil teórica superior a los 10 años por ciclo de batería, la del dispositivo LW004-CT no viene especificada y lo único que sabemos es que es una batería recargable.
- El tamaño: El tamaño es ligeramente inferior al dispositivo Guppy, pero nuestra solución propuesta seguiría siendo más pequeña.
- Iniciativa de alerta por parte del usuario: Ambas soluciones tienen este punto pero la del dispositivo LW004-CT sería una función secundaria.

La tabla resultante de las diferencias es: 3.3.

Dispositivo	Batería	Tamaño	Iniciativa usuario
LW004-PB	-	inferior a Guppy	SI
Propio	10 años	moneda	SI

Tabla 3.3: Tabla diferencias LW004-CT.

3.3 AlertCops

AlertCops es una aplicación móvil desarrollada por el ministerio de interior de España. Se basa en un servicio de alertas de seguridad ciudadanas 3.3 que permite comunicarse directamente de una manera rápida y gratuita con las Fuerzas y Cuerpos de Seguridad del Estado. Además, sirve también como canal de avisos por parte las Fuerzas y Cuerpos de Seguridad del Estado ya que teniendo en cuenta tu ubicación te alertaría de algún suceso importante que estuviese ocurriendo en esa zona. Estas alertas, además de ir acompañadas de la ubicación, pueden sumárseles imágenes y vídeos.

Esta aplicación, a pesar de ser muy completa, tiene el gran problema en su dependencia total del teléfono móvil y su GPS. Esto le quita cierta libertad al usuario ya que tiene que estar pendiente de llevar el smartphone en todo momento incluso al salir a correr. Por el contrario, el dispositivo que planteamos nos ofrece esa libertad añadiéndole el no tener que estar pendiente de la duración de la batería en años.



Figura 3.3: Captura de la aplicación AlertCops

Análisis y diseño

EN este capítulo mencionaremos como vamos a desarrollar nuestra solución. También analizaremos todo lo que necesitaremos para llevarla a cabo para después diseñar la solución que implementaremos.

4.1 Análisis y diseño del proyecto

Nuestra solución se basa en un dispositivo de pequeño tamaño que tras la interacción con el usuario enviará unos datos (alerta) mediante LoRa a un servicio web que determinará la acción a realizar y la llevará a cabo. A partir de aquí extraeremos los requisitos y diseñaremos un producto que los cumplan.

4.1.1 Requisitos

Tras el estudiar la secuencia de acciones que se llevarán a cabo en nuestra solución podemos identificar y extraer una serie de requisitos que debe cumplir al finalizar el desarrollo. Estos se dividen en dos grupos:

- Funcionales:
 - Localización del dispositivo sin GPS: poder ubicar y conseguir la ubicación del dispositivo sin la necesidad de utilizar un GPS.
 - Crear, modificar y eliminar una secuencia: debe dar la posibilidad al usuario de poder realizar todas las tareas relacionadas con la secuencia (crearla, modificarla y eliminarla).
 - Realizar una acción preconfigurada: el sistema tiene que llevar a cabo una acción que a configurado con anterioridad en usuario en el sistema.
 - Detener la secuencia de acciones tras su activación: tras la activación de la secuencia, el sistema debe permitir al usuario detenerla de manera simple.

- No funcionales:
 - Interfaz intuitiva: el resultado final de la interfaz debe ser simple y que facilite su uso sin necesidad de un manual.
 - Diseño muy compacto: el dispositivo resultante debe ser lo más pequeño posible intentando alcanzar el tamaño teórico que hemos estipulado (tamaño similar al de una moneda de dos euros).
 - Bajo coste: se debe conseguir un dispositivo que sea asequible para el mayor público posible.

4.1.2 Análisis

Después de analizar los requisitos identificados en el apartado anterior para nuestra solución, obtenemos dos casos de uso muy importantes y esenciales.

El primero es la definición de la secuencia 4.1.

El segundo es la introducción de la secuencia 4.2.

Caso de Uso	Definición de la secuencia				CU-01
Actores	(I) Usuario Sistema				
Tipo	Primario Esencial				
Referencias	R-001				
Precondición					
Postcondición	La secuencia debe estar guardada en el sistema				
Autor	Iván	Fecha	01/08/20	Versión	1.0
Propósito					
Define una secuencia en el sistema					
Resumen					
El usuario definirá una secuencia que introducirá mediante el pulsador para que el sistema la guarde en un registro para fines futuros.					
Curso Normal					
1	El usuario inicia el proceso de guardado de una secuencia desde el aplicativo.				
		2	El sistema notifica al usuario que el proceso ha iniciado.		
		3	El sistema enuncia una pequeña ayuda que debe seguir el usuario.		
4	El usuario continúa el proceso tras leer la ayuda.				
		5	El sistema emite una señal para indicar al usuario que puede introducir la secuencia.		
6	El usuario introduce la secuencia deseada.				
		7	El sistema pregunta si la secuencia es correcta.		
8	El usuario responde que la secuencia es correcta.				
		9	El sistema guarda la secuencia en el registro correctamente.		
Cursos Alternos					
7b	Si el usuario no ha introducido ninguna secuencia se vuelve al paso 5.				
8b	Si el usuario contesta que la secuencia no es correcta, el sistema ignora la secuencia introducida y se vuelve al paso 5.				
Otros datos					
Frecuencia esperada	1 vez al mes	Rendimiento	Medio		
Importancia	Vital	Urgencia	Alta		
Estado	Pendiente revisión	Estabilidad	Alta		
Comentarios					
Durante todo el proceso habrá la posibilidad de volver al paso 3 de la secuencia normal.					

Figura 4.1: CU01 - Definición de la secuencia.

Caso de Uso	Introducción de la secuencia				CU-02
Actores	(I) Usuario Sistema				
Tipo	Primario Esencial				
Referencias	CU-01, R-002				
Precondición	Debe tener guardada previamente la secuencia a introducir				
Postcondición	El sistema envía durante un periodo de tiempo la ubicación en tiempo real				
Autor	Iván	Fecha	01/08/20	Versión	1.0
Propósito					
Introducción de una secuencia mediante el pulsador para llevar a cabo una acción.					
Resumen					
El usuario presionará el pulsador del dispositivo realizando la secuencia escogida previamente por él en la aplicación. Tras esto, el sistema realizará la acción que corresponde a dicha secuencia según especifique el registro del usuario en la aplicación web.					
Curso Normal					
1	El usuario introduce la secuencia deseada mediante el pulsador.				
		2	El sistema comprueba si la secuencia introducida es la correcta.		
		3	Envía al servidor una notificación de secuencia correcta.		
		4	El servidor pide los datos de los sensores disponibles y el sistema se los proporciona.		
		5	La aplicación accede al registro del usuario y realiza la acción definida por este para la secuencia introducida.		
Cursos Alternos					
2b	El sistema comprueba la secuencia y no corresponde con ninguna del registro. Se debe volver al punto 1 de la secuencia normal.				
3b	El sistema no puede contactar con el servidor debido a falta de conexión o cobertura. El sistema debe esperar a entrar en una zona con cobertura para continuar con el proceso.				
5b	No existe una acción a realizar con dicha secuencia en el registro del usuario. El sistema no realiza ninguna acción y finaliza el CU.				
Otros datos					
Frecuencia esperada	1 vez al día	Rendimiento	Alto		
Importancia	Vital	Urgencia	Alta		
Estado	Pendiente revisión	Estabilidad	Moderada		
Comentarios					
Los datos de los sensores del sistema se obtienen en segundo plano cada x tiempo					

Figura 4.2: CU02 - Introducción de la secuencia.

A mayores de estos dos, también identificamos otros casos de uso más simples, que re-

quieren de una o dos acciones por lo que nos limitaremos a mencionarlos. Estos son:

- Creación de un usuario
- Modificación de la secuencia
- Eliminación de la secuencia
- Detención de la secuencia
- Asignación de una acción a realizar con una secuencia determinada

4.1.3 Diseño del proyecto

Para facilitar la realización del diseño conforme a los requisitos descritos con anterioridad, lo dividiremos en 2 partes:

- Parte offline: esta parte se centrará en el dispositivo hardware y su interacción con el usuario.
- Parte online: se ocupará de todo lo relacionado con el servicio web y los procesos necesarios en la nube para asegurar el correcto funcionamiento de la solución.

4.1.3.1 Parte offline

En esta parte debemos tener claro que hardware queremos usar para nuestra solución final pero teniendo en cuenta que debemos realizar un prototipo funcional. Por esto, y tras sondear el mercado hardware, nos decidimos por separar en dos conjuntos el hardware de la solución.

El primer conjunto sería el seleccionado para llevar a cabo el prototipo de la solución. Este estaría compuesto por:

- Un módulo de desarrollo WiFi LoRa 32 KIT v2 fabricado por Heltec: este módulo cuenta con:
 - Microprocesador ESP32, para ser exactos, Tensilica LX6 dual-core de 240MHz.
 - Chipset LoRa SX1276 que admite conexión punto a punto y LoRaWAN.
 - Display OLED
 - Una antena metálica integrada para comunicaciones WiFi y Bluetooth.
 - Una antena externa para comunicaciones LoRa.
- Un botón

A mayores ofrece una librería propia en Arduino para dar soporte al usuario que le facilita la interacción con el display OLED y las acciones con los distintos tipos de comunicaciones disponibles. Este primer conjunto 4.3 4.4 tendría un coste total de 19€ (varía unos céntimos debido al redondeo y la adquisición del botón).

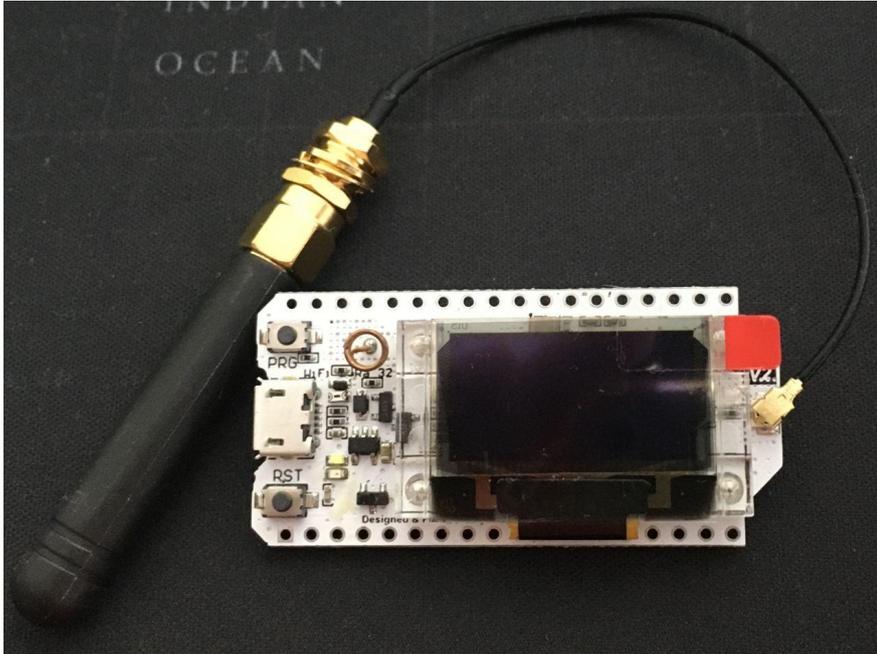


Figura 4.3: Módulo WiFi LoRa 32 KIT v2. Parte frontal.

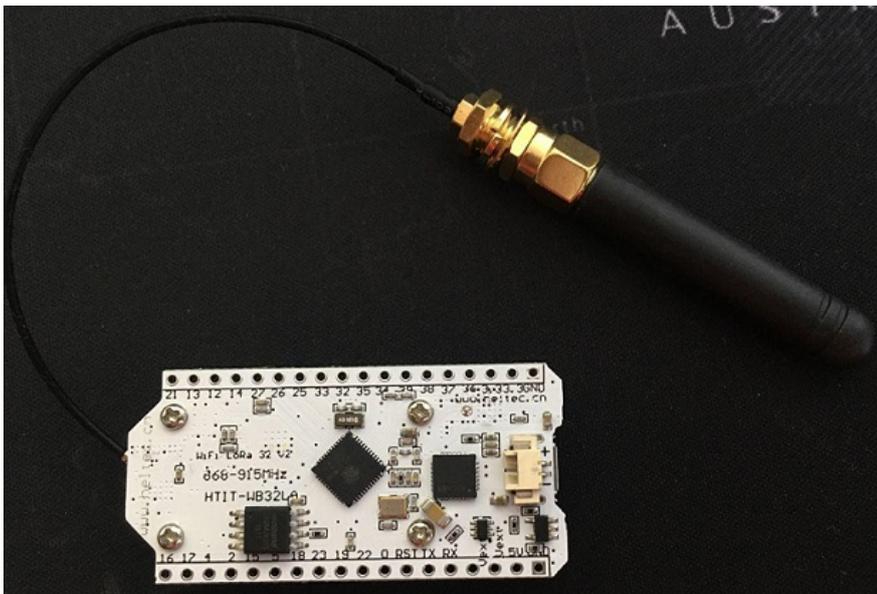


Figura 4.4: Módulo WiFi LoRa 32 KIT v2. Parte trasera.

El segundo conjunto se utilizaría para realizar la versión final de la solución. Este estaría compuesto por:

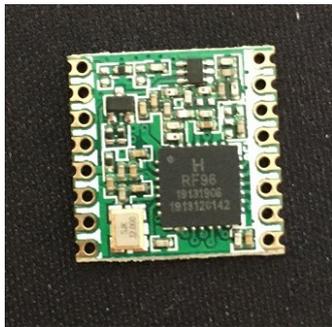
- Un módulo de desarrollo NodeMcu V3 Lua WiFi: este módulo cuenta con:
 - Microprocesador ESP8266 (ESP-12E).
 - Una antena metálica integrada para comunicaciones WiFi.
- Módulo transceptor inalámbrico RFM95W con un chipset LoRa SX1276.
- Un botón.
- Antena auxiliar para conectar a RFM95W.

Este conjunto al tener el módulo de desarrollo de una plataforma de código abierto como es NodeMcu, permite abaratar mucho el precio final. El precio para el módulo de desarrollo 4.5 es de 3€, mientras que para el módulo RFM95W 4.6 es de 7€, dejando el coste total del conjunto en torno a los 10€ (varía unos céntimos debido al redondeo y la adquisición o fabricación de una antena y el botón).

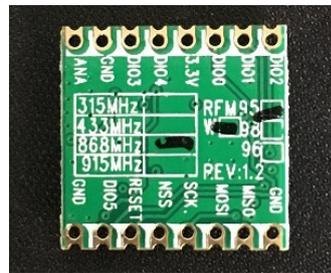


Figura 4.5: Módulo NodeMcu V3 Lua WiFi.

La razón de porque hemos decidido realizar una diferenciación del hardware es debido a que durante la fase de pruebas el primer conjunto nos aporta mayor información del estado del sistema gracias a que podemos mostrar las acciones realizadas por el sistema en todo momento a través del display OLED añadiendo unas instrucciones en el código. Por su contra, el segundo conjunto, para poder visualizar el estado del sistema tendríamos que adquirir, instalar y programar un display con los inconvenientes que acarrea, como una mayor ocupación de pines en el módulo de desarrollo, un mayor número de instrucciones o la aparición de errores derivados del display.



(a) Parte frontal.



(b) Parte trasera.

Figura 4.6: Módulo RFM95W.

A pesar de esta diferencia a la hora de desarrollar el proyecto, el sistema final una vez esté implementado y funcional será compatible con el segundo conjunto a expensas de un cambio en el código en la instrucción de asignación de pines.

Además del hardware, en este apartado contaremos con el software encargado de gestionar las secuencias, tanto para crearlas, modificarlas y eliminarlas, además de controlar su correcta introducción. Este software contará con una interfaz que simplificará y facilitará la gestión de las secuencias al usuario.

4.1.3.2 Parte online

En el diseño de esta parte debemos tener en cuenta que el dispositivo que debe conectar a un servicio web a través de LoRa. Para poder realizar esa conexión, vamos a necesitar trabajar con el protocolo LoRaWAN.

Para trabajar con LoRaWAN se necesita, además del dispositivo final, dos componentes clave para asegurar su correcto funcionamiento:

- Un gateway: que recoja la señal emitida por el dispositivo del usuario.
- Un servidor de red: que trate la información recibida en el gateway y la transmita al servicio web deseado.

La arquitectura típica de una red LoRaWAN sería la mostrada en la imagen 4.7.

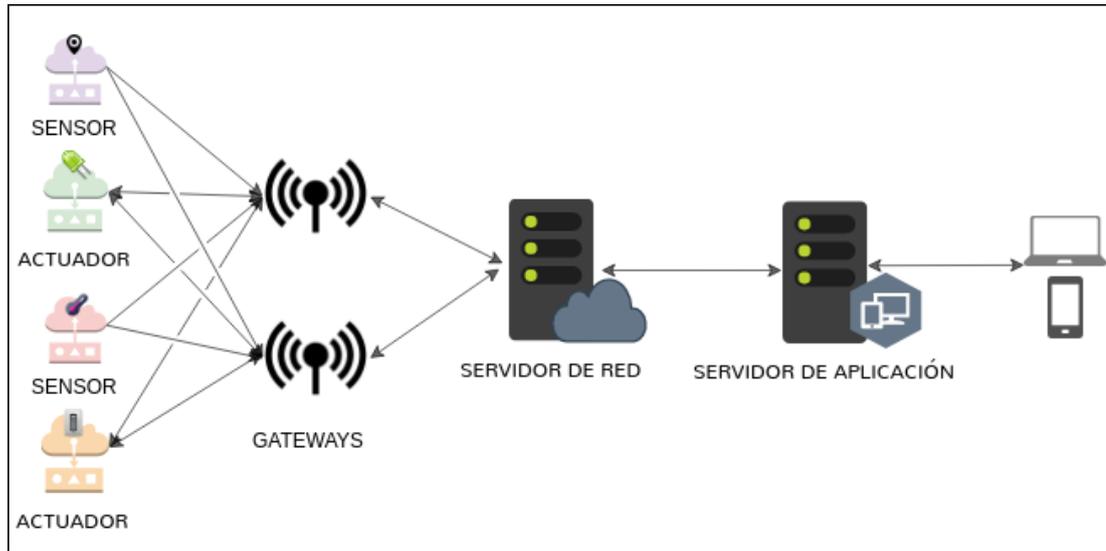


Figura 4.7: Arquitectura típica de una red LoRaWAN.

Como la finalidad de este proyecto es conseguir una solución lo más barata posible cumpliendo las funcionalidades deseadas utilizaremos la red de The Things Network [36]. The Things Network (TTN) es una iniciativa creada por Wienke Giezeman en 2015 que se cimienta en la comunidad para establecer una red global gratuita y abierta de IoT. En la actualidad, está presente en 150 países, con más de 12000 gateways abiertos y con una comunidad en constante crecimiento superando los 115000 miembros activos. Esta iniciativa tiene mayor presencia en Europa llegando a brindar cobertura total a países completos como los Países Bajos o Suíza y más del 65 % del territorio como Alemania o Bélgica.

The Things Network (TTN) tiene un funcionamiento simple. La comunidad aporta los gateways para que tu dispositivo se conecte a la red y un aplicativo online para visualizar y procesar la información que ha recibido de tu dispositivo. En cuanto al aplicativo online, para poder acceder, primero debes registrarte gratuitamente en su página web [36] y posteriormente acceder al apartado de consola. En este punto se te mostrará una pantalla como la de la imagen 4.8. En ella podemos observar como TTN distingue entre dos categorías:

- Aplicaciones: es el lugar donde aparecen las aplicaciones que crea el usuario para recibir los datos enviados por su dispositivo y posteriormente tratarlos, procesarlos y enviarlos al servicio web que se desee mediante diferentes opciones y protocolos.
- Gateways: es el apartado donde aparecerían los gateways que nosotros aportamos a la red y que tenemos control sobre ellos.

En nuestro caso, sólo necesitamos el apartado de aplicaciones. En el apartado 5.2 mostraremos como crear una aplicación destinada a nuestro proyecto junto con la conexión de nuestro dispositivo y el tratamiento/procesado de los datos recibidos.

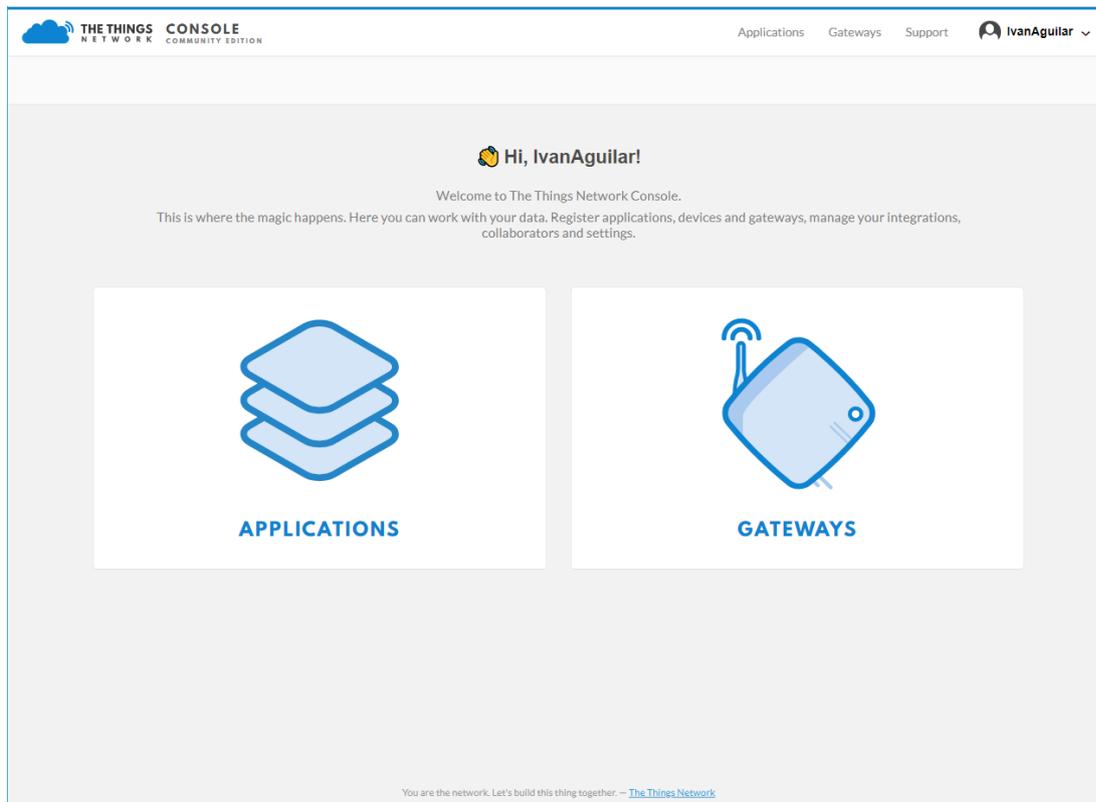


Figura 4.8: Consola TTN.

En cuanto al servicio web para realizar una acción concreta utilizaremos la plataforma IFTTT [18]. Esta plataforma nos brinda un sinfín de opciones que nos permiten conectar múltiples aplicaciones entre sí con la finalidad de crear y programar acciones que responden a eventos surgidos en la aplicación origen. Para entenderlo mejor, un ejemplo sería que cada vez que alguien te etiquete en una foto en Facebook, te notificara de manera automática a Telegram con un texto personalizado. Para poder gozar de esta plataforma es necesario registrarse de manera gratuita en su página web [18]. Posteriormente en el apartado 5.3 explicaremos la creación desde cero de una acción para utilizar en nuestro proyecto.

4.1.3.3 Diseño final

Teniendo en cuenta ambas partes del diseño y los requisitos que debemos cumplir podemos crear un esquema básico de como sería nuestro sistema con todas las partes ensambladas. En la imagen 4.9 podemos observar claramente las 3 partes principales del sistema:

- El dispositivo junto con el aplicativo offline: es la pareja formada por el dispositivo y el software encargado de procesar la secuencia introducida para posteriormente mandar la información necesaria al aplicativo interno.

- **Aplicativo interno:** esta parte del sistema es la que se encuentra en TTN y se encarga de recibir y procesar los datos enviados desde el dispositivo.
- **Aplicativo online:** es el servicio web IFTTT que se encargará de realizar la acción configurada, además de recibir los datos del aplicativo interno.

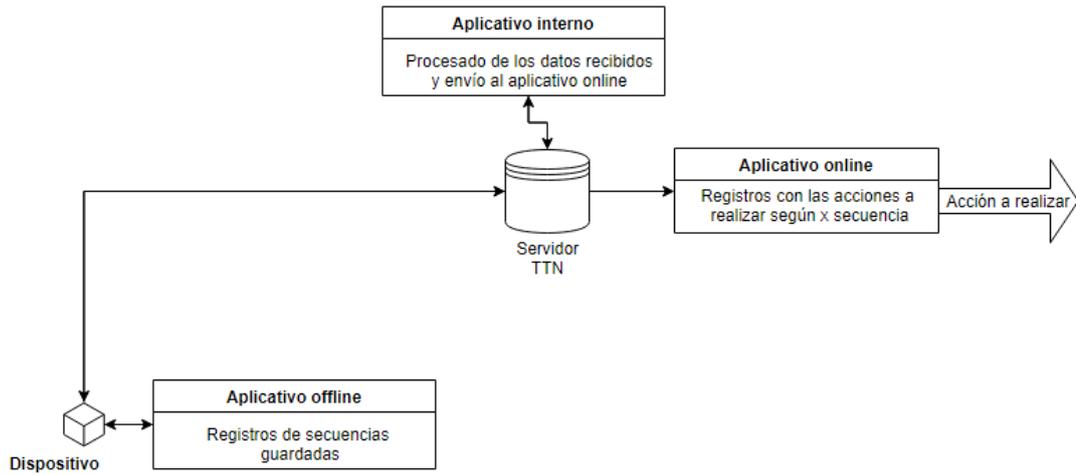


Figura 4.9: Esquema básico del sistema.

Implementación e integración

EN este capítulo explicaremos los pasos seguidos para realizar la implementación e integración del sistema dividiéndolo en las tres partes del mismo:

- Aplicativo offline
- Aplicativo interno
- Aplicativo online

5.1 Aplicativo offline

En esta parte del sistema es necesario la realización del programa Arduino que se ejecutará en el módulo de heltec WiFi LoRa 32 Kit v2. Además, se prooverá al usuario de una interfaz gráfica que facilitará el registro de la secuencia.

5.1.1 Hardware

Acompañando al módulo heltec WiFi LoRa 32 Kit v2 tendremos un simple circuito de un pulsador y una resistencia. Para poder realizar el circuito con éxito es necesario conocer el esquema de pines del módulo. En la imagen 5.1 observamos la asignación de los pines del módulo según el esquema proporcionado por el fabricante Heltec. Con esto, podemos diferenciar tanto los pines que van ser usados para la conexión de LoRa, los de alimentación/tierra y los pines analógicos como los pines digitales del cual usaremos uno de estos pines para la conexión con el pulsador.

Teniendo esto en cuenta, conectaremos al circuito por el pin 13 un pulsador con una resistencia pull-down de $10k\Omega$ (ohmios) 5.2.

Tras verificar el esquema y el pin de conexión, llevamos a cabo el circuito 5.3.

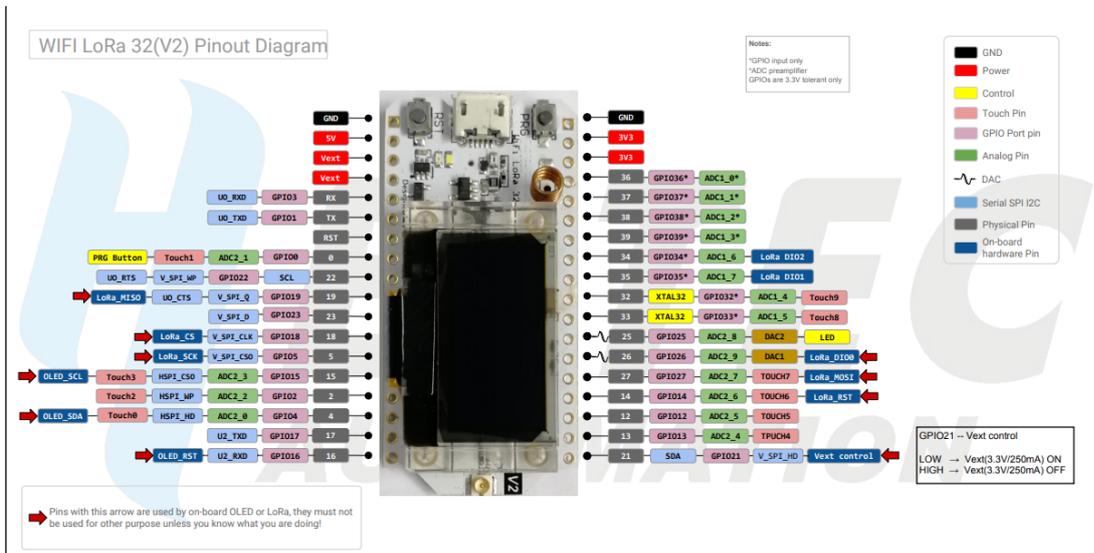


Figura 5.1: Diagrama de pines de WiFi LoRa 32 Kit v2.

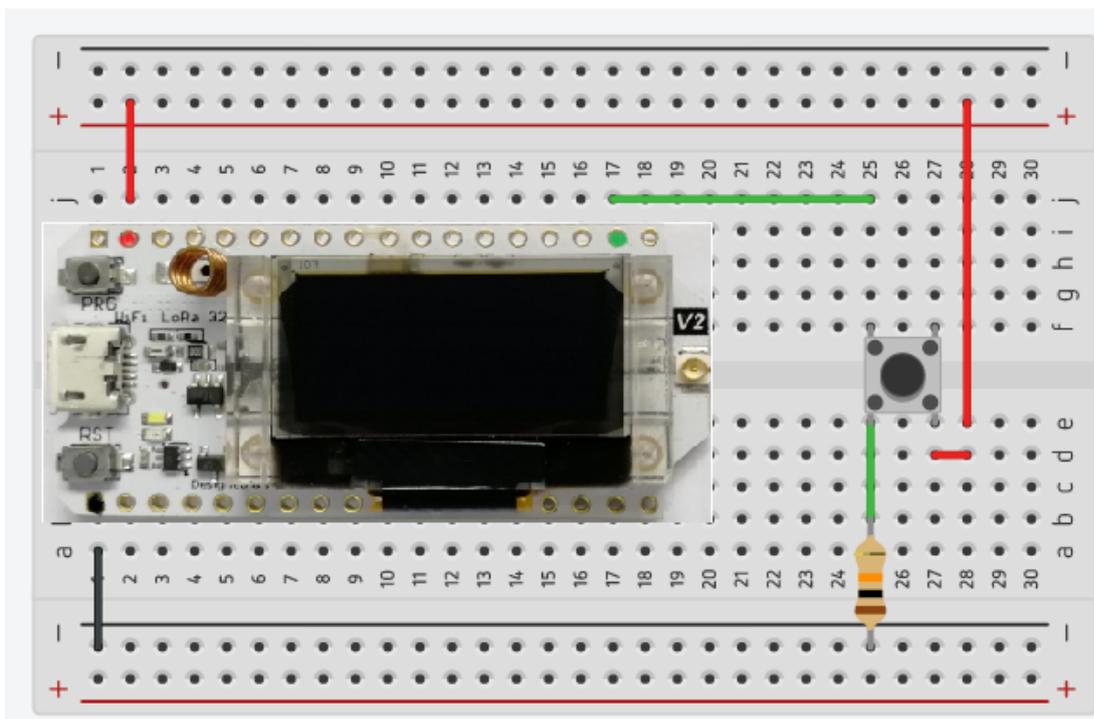


Figura 5.2: Circuito esquemático.

5.1.2 Programa

Con el circuito montado toca desarrollar el programa Arduino que controlará el correcto funcionamiento del módulo y las secuencias. Antes de nada, para poder trabajar con el módulo

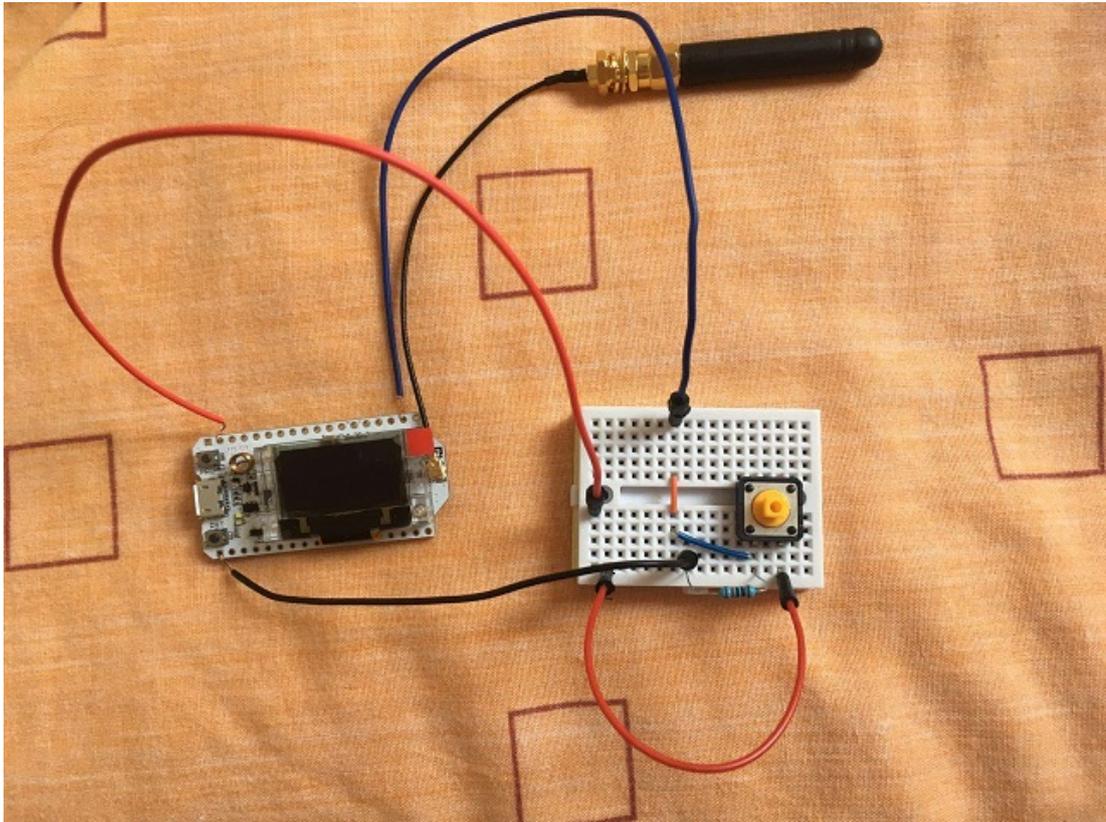


Figura 5.3: Circuito.

de desarrollo WiFi LoRa 32 Kit v2 debemos preparar el IDE de Arduino. Para ello, seguiremos estos pasos:

- Primero: instalaremos el gestor de tarjetas de Heltec tal como se ve en la imagen 5.4, en el campo gestor de URLs adicionales de tarjetas.
- Segundo: añadiremos la librería de desarrollo de Heltec para ESP32 5.5.
- Tercero: seleccionar la tarjeta correspondiente a nuestro módulo 5.6a.
- Cuarto: asegurarse de que la región de la configuración de la tarjeta se encuentra en Europa (868 MHz) En la imagen 5.6b observamos que corresponde con el campo LoRa-WAN Region.

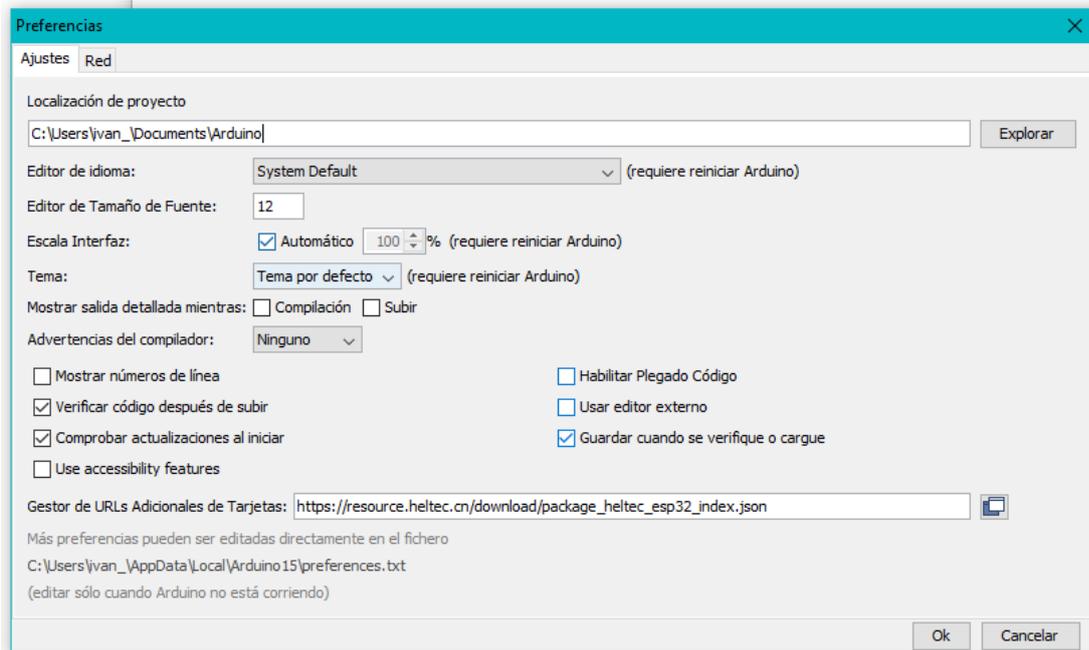


Figura 5.4: Gestor de tarjetas.

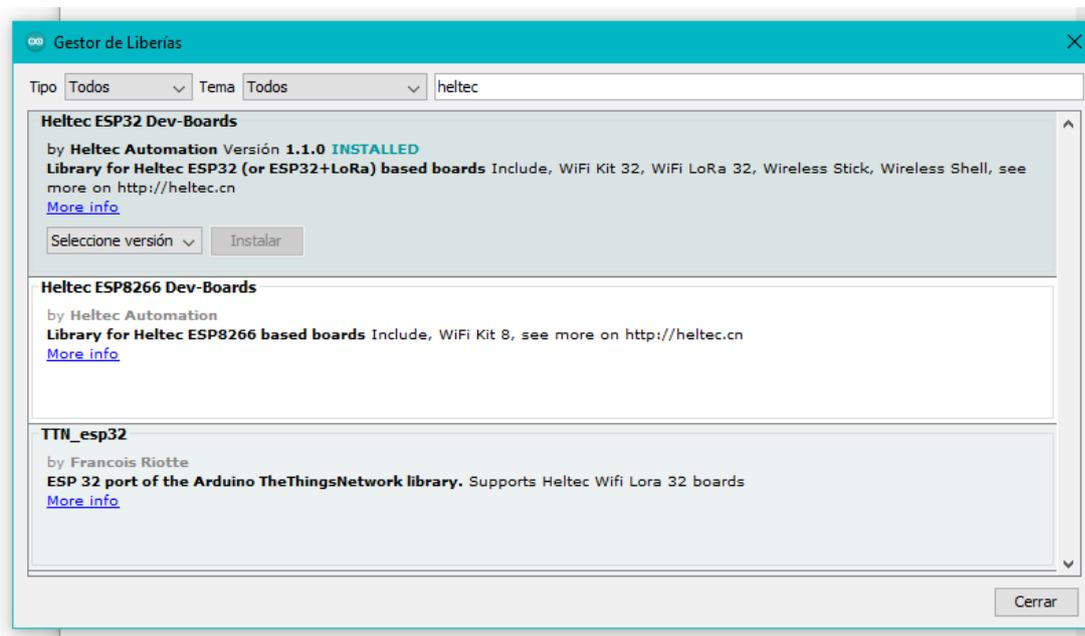
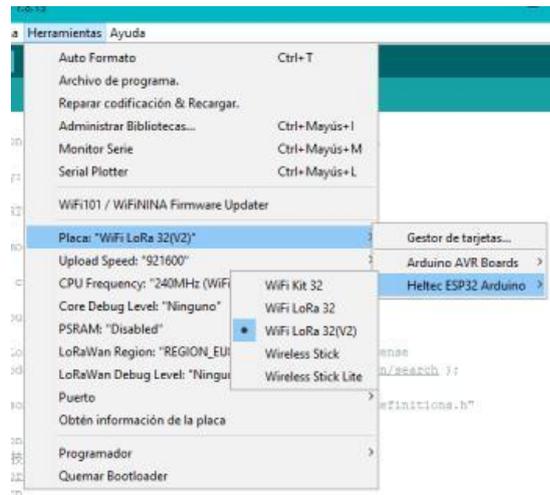


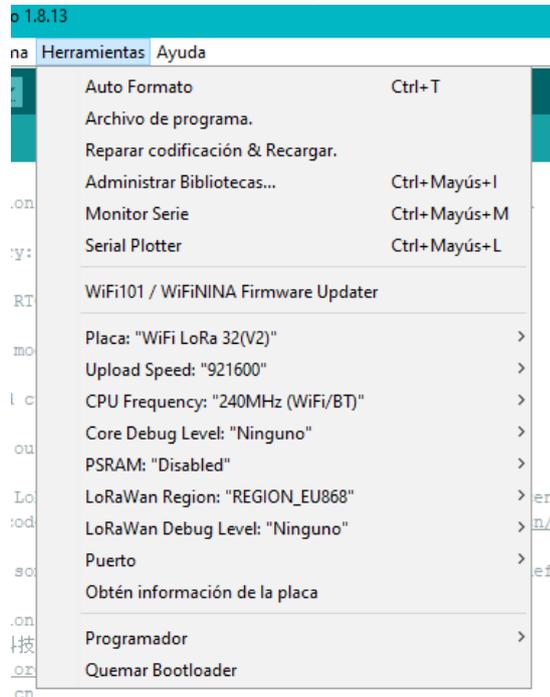
Figura 5.5: Librería Heltec ESP32.

Con esto hecho, debemos tener en cuenta de que manera nos conectaremos con LoRa-WAN, mediante OTAA o ABP.

OTAA (Over-The-Air-Activation) es la manera más recomendada de conectarnos y su



(a) Selección tarjeta Heltec.



(b) Configuración básica.

principal ventaja es la seguridad ya que la sesión se crea en cada conexión y se renueva cada vez que se pierde la conexión, se apaga o se reinicia el dispositivo dificultando así la posibilidad de un robo de sesión y clonación del dispositivo. Sus parámetros de configuración son:

- DevEUI: es el identificador único de cada dispositivo.
- AppEUI: es el identificador único de la aplicación.

- **AppKey:** es la clave secreta cifrada con AES de 128 bits y es compartida entre el dispositivo y la red.

La secuencia de la conexión con OTAA es la siguiente:

- El dispositivo solicita un inicio de sesión a la red con los datos de configuración y se abre una ventana de recepción.
- El gateway recibe la solicitud y la reenvía al servidor.
- El servidor contrasta y verifica los datos recibidos.
- Si son correctos (en caso contrario se rechaza la solicitud) se le asigna una sesión temporal y se la envía al dispositivo por medio del gateway.
- El dispositivo recibe la sesión temporal y puede proceder a enviar datos a la red.

Por su parte ABP (Activation By Personalization) es el modo más sencillo de conectarnos debido a que no requiere de solicitud de inicio de sesión para enviar datos. Esto es por la sesión ya está asignada manualmente lo que es muy práctico para dispositivos en movimiento o con dificultades de recepción. Los parámetros de configuración de este modo son:

- **DevAddress:** es la dirección lógica que se utilizará para la comunicación.
- **NetworkSessionKey:** es la clave de cifrado entre el dispositivo y el operador que se utilizará en las comunicaciones y que permitirá verificar la integridad de los mensajes.
- **ApplicationSessionKey:** es la clave de cifrado entre el dispositivo y el operador, a través de la aplicación, que se utilizará en las comunicaciones y que permitirá validar la integridad de los mensajes.

La secuencia de la conexión con ABP es la siguiente:

- El dispositivo envía los datos deseados a la red con los datos de configuración.
- El gateway recibe los datos y valida que los parámetros de configuración se correspondan a la sesión.
- Si es correcta la sesión procesa los datos, sino se rechazan.

Todos los parámetros de configuración mencionados anteriormente los obtenemos de la consola de TTN que explicaremos en la siguiente sección.

Observando las diferencias entre OTAA y ABP, decidimos realizar la conexión mediante ABP. La razón es porque nuestro dispositivo no está destinado para estar en un lugar fijo sino que se encontrará en movimiento la mayor parte del tiempo.

Con todo esto en cuenta podemos programar el código que controlará nuestro módulo y las secuencias. En dicho código hay dos partes importantes que debemos tener muy presente:

- Controlar la secuencia: es la parte encargada de determinar si las pulsaciones introducidas corresponden con la secuencia guardada .
- La asignación de los datos de configuración de ABP: son las variables donde se registran los valores correspondientes a la configuración de los valores para la conexión con LoRaWAN mediante ABP.

El algoritmo encargado de controlar la secuencia tanto se utilizará para cambiar el estado principal del sistema, tanto para cambiar a alerta como para volver a reposo. Es decir, servirá para activar la alerta en una primera instancia y para desactivarla posteriormente. Este algoritmo funciona de la siguiente manera:

- El sistema parte de un estado de reposo o de alerta del cual sale tras recibir una primera pulsación.
- En ese momento se pasa a una función auxiliar que esperará por la siguiente pulsación. Si no se produce una pulsación en menos de 15 segundos la validación de la secuencia se cancela y vuelve al estado original.
- Cuando el sistema reciba la siguiente pulsación, medirá el tiempo transcurrido entre ambas para determinar si es una pulsación corta o larga. Será corta si ha transcurrido menos de 3 segundo y medio, en caso contrario será larga.
- El proceso se repetirá hasta conseguir las 4 pulsaciones requeridas para la secuencia.
- Una vez se tengan las 4 pulsaciones se comprueba su validez comparandolas con el registro de secuencias guardadas. Si es correcta el sistema cambiará su estado principal y en caso contrario volverá a su estado original.

EL diagrama de estado del algoritmo sería el de la imagen 5.7.

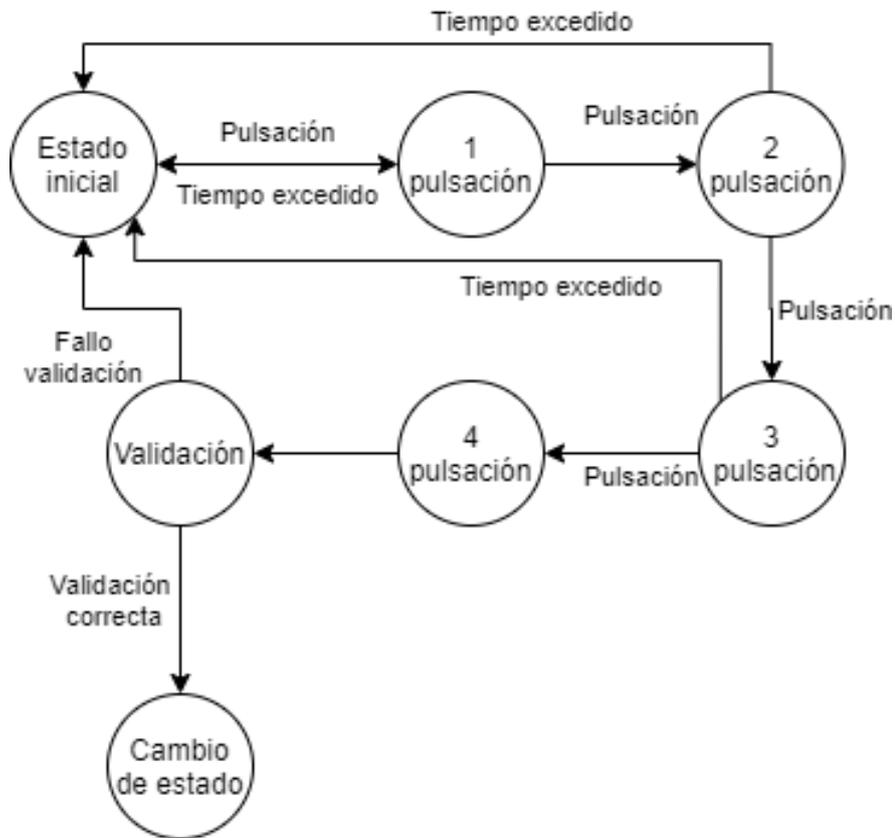


Figura 5.7: Diagrama de estado del algoritmo.

5.1.3 Interfaz gráfica

Para facilitar al usuario el registro y manejo de las secuencias se ha creado una interfaz gráfica simple e intuitiva. Este prototipo se ha programado en Python utilizando el módulo PySimpleGUI que nos agiliza dicha creación. Este módulo distribuye el espacio disponible en filas y columnas, permitiendo la colocación de los objetos gráficos acorde a nuestro gusto mediante una posición definida por coordenadas fila-columna.

La interfaz desarrollada consta de un menú desplegable para que escogamos que secuencia queremos guardar (secuencia 1, secuencia 2, secuencia 3), 4 menús spin para escoger entre pulsación corta y pulsación larga y dos botones, uno de guardado y otro para cancelar y cerrar la interfaz. Una secuencia normal de uso de la interfaz sería:

- Inicialmente al abrir la interfaz 5.8.
- Tras seleccionar una secuencia y sus pulsaciones antes de guardar 5.9.
- Después de pulsar Submit y guardar la secuencia 5.10.

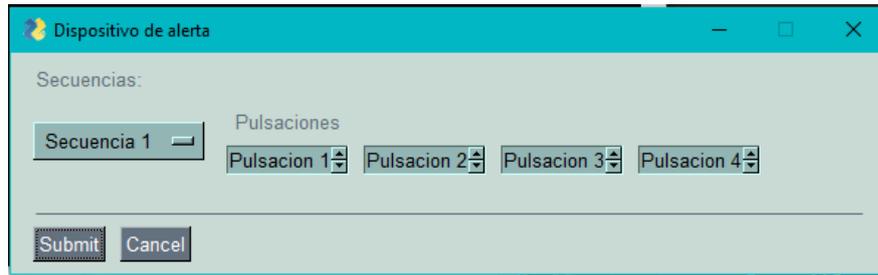


Figura 5.8: Interfaz prototipo principal.



Figura 5.9: Interfaz prototipo principal 2.

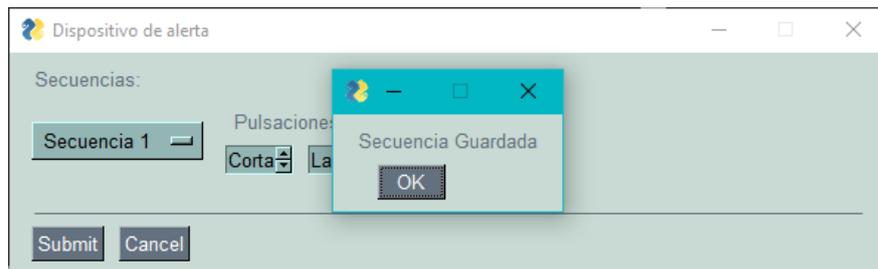


Figura 5.10: Interfaz prototipo Pop up.

5.2 Aplicativo interno

Esta sección del sistema es la correspondiente a la recepción de los datos enviados desde el aplicativo offline a través de LoRaWAN y su procesamiento para su posterior envío al aplicativo online. Para eso, trabajaremos desde la consola que nos proporciona la plataforma The Things Network (TTN).

Lo primero que debemos hacer es crear una aplicación que recepcionará los datos enviados por nuestro dispositivo. Tal como nos muestra la imagen 5.11, en este punto sólo tendríamos que determinar el nombre que le queremos dar y una pequeña descripción si la requiere, ya que el resto de campos los generará automáticamente la plataforma.

El siguiente punto que debemos hacer es registrar el dispositivo en la plataforma y asociarlo a nuestra aplicación (la que acabamos de crear). Para ello, debemos dirigirnos a la pestaña

ADD APPLICATION

Application ID
The unique identifier of your application on the network

alerta-localizador

Description
A human readable description of your new app

Eg. My sensor network application

Application EUI
An application EUI will be issued for The Things Network block for convenience, you can add your own in the application settings page.

EUI issued by The Things Network

Handler registration
Select the handler you want to register this application to

ttn-handler-eu

Cancel Add application

Figura 5.11: TTN Crear aplicación.

de dispositivos y seleccionar registrar. Esto nos mostrará una pantalla como la de la imagen 5.12. Nuevamente, todos los campos se generarán automáticamente excepto el nombre del dispositivo. Tras aceptar, nos mostrará la pantalla de configuración y ajustes del dispositivo donde aparecerán todos los datos de configuración necesarios tanto para la conexión OTAA como la conexión ABP 5.13.

Con estos pasos, ya tendríamos una aplicación que recibe datos de un dispositivo LoRa. El problema es que esos datos no se guardarían y se desecharían en tiempo real. La solución es añadir un almacenamiento de datos desde la pestaña de integraciones 5.14. Esta integración nos permitirá guardar nuestros datos durante 7 días de manera gratuita.

Actualmente nuestro sistema ya tiene conectado el aplicativo offline y el interno pero le falta la conexión entre el aplicativo interno y el aplicativo online, además de tener geolocalizado el dispositivo. Lo primero que solucionaremos será la geolocalización. Para localizar nuestro dispositivo de manera eficiente y sin GPS utilizaremos el servicio gratuito de geolocalización de LoRa Cloud [37]. Este servicio permite localizar un dispositivo LoRa sin necesidad de incorporar hardware adicional y con una precisión de 50 metros, menor precisión frente al GPS, incluyendo los interiores de los edificios. Además, este servicio no implica gasto adicional en el término de energía ya que utiliza la infraestructura de la red y los términos RSSI (Received Signal Strength Indicator), SNR (signal-to-noise ratio) y TOA (time of arrival) de los metadatos de la trama enviada para localizar el dispositivo.

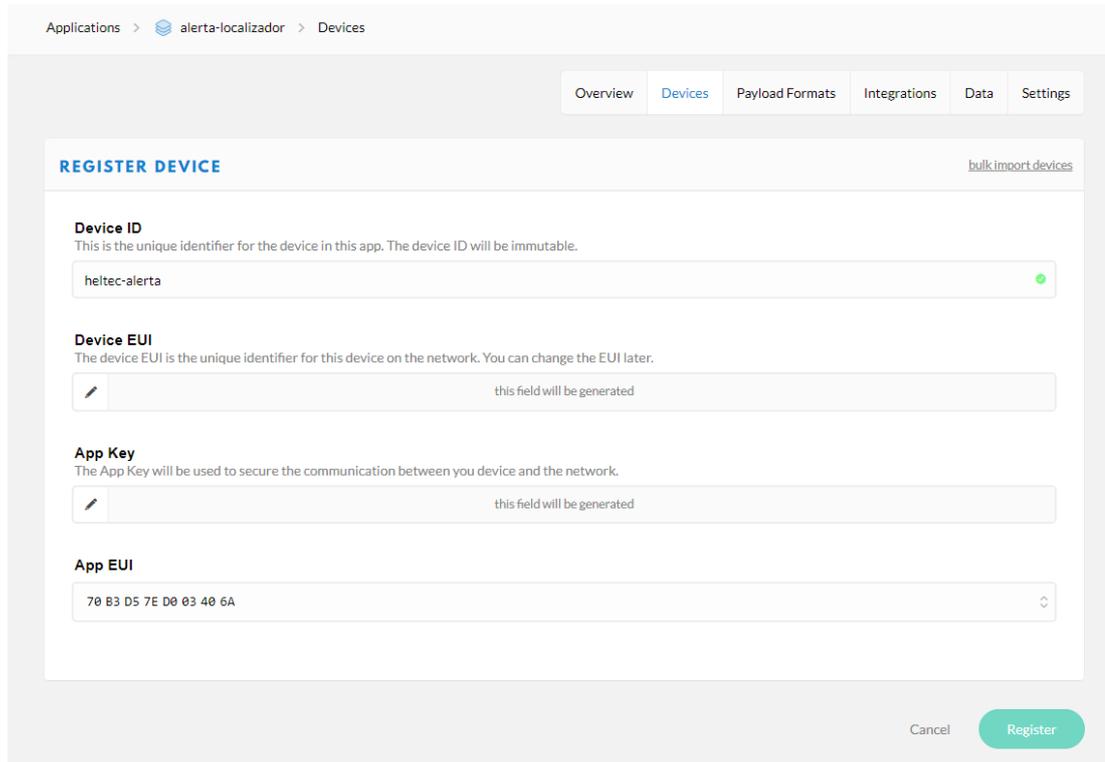


Figura 5.12: TTN Registrar dispositivo.

Para poder incorporarlo a nuestro sistema, utilizaremos Collos, una integración de la plataforma TTN que es la fase beta del servicio LoRa Cloud. Para ello, añadiremos la integración y en la pantalla que nos muestra 5.15 deberemos rellenar los datos:

- Process ID: el nombre que le queremos dar para identificar esta integración.
- Access Key: seleccionamos la correspondiente a nuestra app. En nuestro caso la default key.
- URL: Es la dirección url de la API del servicio de geolocalización LoRa Cloud. Hay 3 tipos de api y por tanto 3 urls:
 - RSSI: la api utiliza el parámetro RSSI de la trama para localizar el dispositivo.
 - TDOA: la api utiliza el parámetro RSSI combinado con el parámetro TOA de la trama para localizar el dispositivo.
 - TDOA + WiFi: utiliza el mismo método que TDOA pero añadiendo a la fórmula los datos de puntos de acceso WiFi contenidos en la carga útil de la trama.
- Collos subscription key: es la llave de suscripción de nuestro usuario en Collos (posteriormente LoRa Cloud). Para conseguirla, debemos registrarnos gratuitamente en su

Applications > alerta-localizador > Devices > heltec-alerta

Overview Data Settings

DEVICE OVERVIEW

Application ID **alerta-localizador**

Device ID heltec-alerta

Activation Method ABP

Device EUI <> 00 16 58 0A E5 14 C1 78

Application EUI <> 70 B3 D5 7E D0 03 40 6A

Device Address <> 26 01 10 51

Network Session Key <> [redacted]

App Session Key <> [redacted]

Figura 5.13: TTN Dispositivo.

Overview Devices Payload Formats Integrations Data Settings

ADD INTEGRATION

 **Data Storage** (v2.0.1)
The Things Industries B.V.
Stores data and makes it available through an API. Your data is stored for seven days.

Cancel Add Integration

Figura 5.14: TTN Data Storage.

servicio, seleccionar la suscripción gratuita y dirigirnos al apartado de manejo de tokens. Ahí encontraremos dos llaves, de las cuales podremos utilizar cualquiera de ellas para este campo 5.16.

Como último paso para completar el aplicativo interno nos quedaría la conexión entre este y el aplicativo online. Para ello, se usará la integración IFTTT Maker. Previo a este paso, debemos acondicionar los datos que recibimos desde el dispositivo para volverlo más entendible

Applications > alerta-localizador > Integrations

ADD INTEGRATION



COLLOS
collaborative location service

Collos (v2.7.10)
Semtech Corporation

Collos offers a collection of location-ready APIs all allowing free prototyping with instant sign-up. What is more, within the Collos framework and market place it is easy to build on top of existing technology adding intelligence and learning to make new and powerful location services. Your new location service can be kept private or shared back to Collos for others to use. When sharing back to Collos, you set the price of using it - it can even be free, if you choose.

[documentation](#) [support](#)

Process ID
The unique identifier of the new integration process

Access Key
The app access key

no selection

URL
The URL of the endpoint

Method
The HTTP method to use

POST
✔

Collos subscription key
Find your subscription key in your Collos profile

Figura 5.15: TTN Collos.

y poderse pasar a IFTTT Maker.

Para esto, modificaremos la función decoder del payload formats de la consola de TTN. Hay que tener en cuenta que la función decoder es un script que se ejecuta automáticamente para cada paquete de entrada que recibe nuestra aplicación de la plataforma TTN. El lenguaje de programación de este script es JavaScript. Teniendo esto presente, realizamos un pequeño script que devuelve los datos en formato JSON por los bytes de datos que hemos enviado desde el dispositivo.

Con esto hecho, añadimos la integración IFTTT Maker a nuestra aplicación de TTN. Debemos rellenar los campos que nos solicita la plataforma. Estos son:

- El nombre que le queramos dar al proceso de la integración. En nuestro proyecto lo hemos nombrado alerta.
- EL nombre del evento en IFTTT que recibirá el evento.
- La key que pide es la que nos proporciona IFTTT cuando entramos en webhooks y

The screenshot shows the 'Manage Tokens' interface in the LoRa Cloud Geolocation developer portal. At the top, the LoRa Cloud logo and 'CLOUD' text are visible on the left, and 'DEVELOPER PORTAL | SEMTECH.COM' with a 'LOG OUT' button is on the right. The main header is 'LoRa Cloud™ Geolocation'. The left sidebar contains a navigation menu with 'MANAGE TOKENS' selected. The main content area displays a message about the 'Free tier' and the URI 'https://gls.loracloud.com'. Below this, a table lists two tokens: 'Primary Token' and 'Secondary Token'. Each token entry includes a 'Rename' button, the token name, a long alphanumeric token string, and buttons for 'View', 'Copy', and 'Renew'.

Figura 5.16: TTN LoRa Cloud.

vemos la parte final de la url que nos muestra estando logueados.

- Los valores que le enviaremos.

5.3 Aplicativo online

Este apartado del sistema es el más simple de los tres que lo forman. En él iniciaremos un servicio web en la plataforma IFTTT que realizará una acción configurada por nosotros.

Lo primero será crear una cuenta en la plataforma IFTTT. A continuación, crearemos lo que en la plataforma se denomina una receta. Los pasos son los siguientes:

- En la pantalla principal, pulsamos en la pestaña create.
- En la siguiente pantalla, pinchamos en la palabra THIS que aparece en grande en la frase "IF THIS THEN THAT".
- En el buscador que nos aparece pondremos webhook. Este es el nombre que recibe maker en su versión mejorada en la plataforma IFTTT.

- Luego seleccionamos el desencadenante (trigger) que nos aparece (receive a web request) y le damos el nombre "alerta".
- Posteriormente, seleccionamos la acción que deseamos pinchando en la palabra THAT que aparece en grande en la frase "IF THIS THEN THAT". En nuestro caso, seleccionamos Telegram.
- Seleccionamos send message y lo configuramos según la imagen 5.17. Esto nos proporcionará que secuencia ha sido pulsada y la ubicación en el momento de la pulsación mediante un mensaje de Telegram.

Todos estos pasos están documentados gráficamente en el anexo Creación de una receta en IFTTT [A.1](#).

Complete action fields

Step 5 of 6

Target chat

Private chat with @IFTTT 

Use the [@IFTTT](#) bot on Telegram to connect new [groups](#) or [channels](#).

Message text

EventName `
`
Secuencia: Value1 ,
[http://www.openstreetmap.org/?mlat= Value2](http://www.openstreetmap.org/?mlat=Value2)
&mlon= Value3 #map=18

Some HTML tags are supported: b, i, a, br, pre, code. 

Include web page preview?

Enabled 

Telegram will check the first URL in the text to include a small preview of the page. Disabled by default.

Create action

Figura 5.17: IFTTT Configuración acción.

Capítulo 6

Pruebas

EN este capítulo mostraremos todas las pruebas realizadas en el sistema para corroborar su correcto funcionamiento.

6.1 Pruebas funcionamiento general

Para comprobar el correcto funcionamiento del sistema se ha llevado a cabo una batería de pruebas. Esta batería ha consistido en la ejecución de la función principal del sistema durante 7 días, 2 veces al día. Esta función siguió los siguientes pasos:

- Introducción de la secuencia registrada en el sistema mediante el pulsador.
- El servidor recibe el mensaje 6.1. De la trama recibida, el servidor recoge el metadata 6.2 para enviárselo a la integración Collos/LoRa Cloud para obtener la localización.
- Con todos los datos procesados, el servidor se los envía a IFTTT.
- IFTTT los recibía y envía el mensaje al Telegram 6.3 con la ubicación obtenida 6.4.

Applications > alerta-localizador > Devices > heltec-alerta > Data

Overview Data Settings

APPLICATION DATA || pause 🗑 clear

Filters: uplink downlink activation ack error

time	counter	port	
09:38:52	0		
09:38:49	0	2	retry confirmed payload: 01 secuencia: 1

Figura 6.1: Captura 1 de las pruebas generales.

```

{
  "secuencia": 1
}

Metadata
{
  "time": "2020-09-03T07:38:49.703310407Z",
  "frequency": 867.3,
  "modulation": "LORA",
  "data_rate": "SF12BW125",
  "coding_rate": "4/5",
  "gateways": [
    {
      "gtw_id": "eui-b827ebffed4975f",
      "timestamp": 1405224780,
      "time": "2020-09-03T07:38:49.679097Z",
      "channel": 4,
      "rssi": -121,
      "snr": -17.5,
      "latitude": 43.36825,
      "longitude": -8.41083,
      "altitude": 40
    }
  ]
}

Estimated Airtime
827.392 ms

```

Figura 6.2: Captura 2 de las pruebas generales.

Estas pruebas nos reportaron un resultado muy satisfactorio teniendo en 12 de 14 pruebas el funcionamiento esperado. La razón por la que las 2 pruebas restantes no funcionaron correctamente es que el servidor europeo de TTN estaba offline ese día.



Figura 6.3: Captura 3 de las pruebas generales.

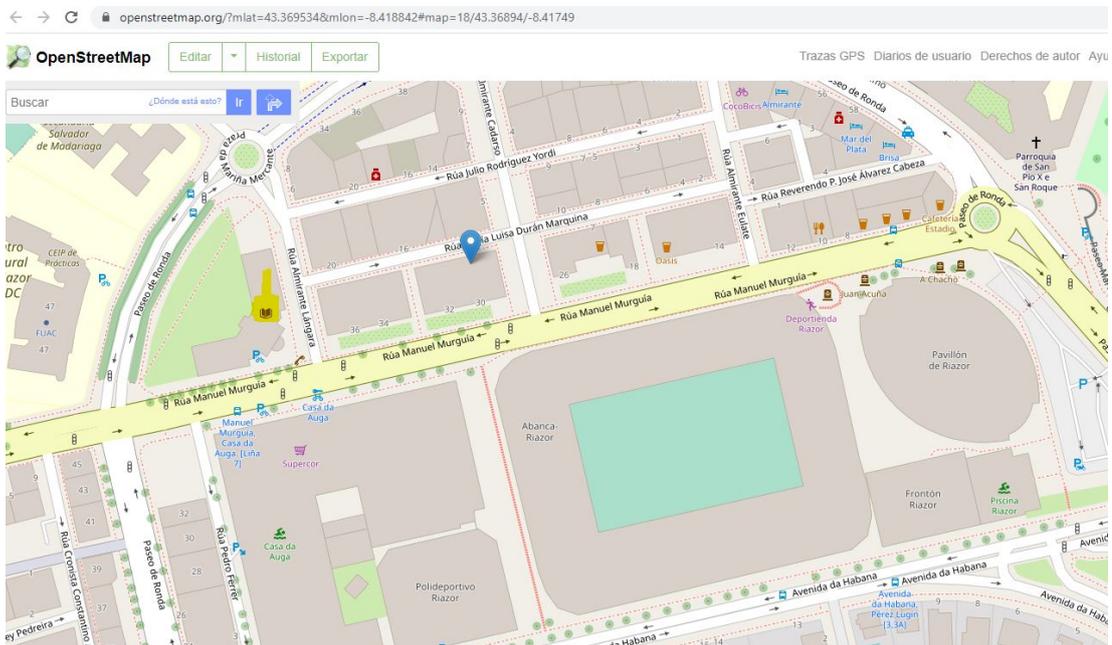


Figura 6.4: Captura 4 de las pruebas generales.

6.2 Pruebas obtención localización

Tras las pruebas generales, no centramos en realizar un pequeño muestreo. Este muestreo constará de 15 alertas enviadas desde la misma localización con el fin de obtener un mapa de todas las localizaciones obtenidas y así comprobar su fiabilidad.

Los datos principales del muestreo son:

- Localización de referencia: Centro Universitario de Riazor - UDC (A Coruña) Rúa Almirante Lángara, s/n, latitud: 43.369393, longitud: -8.420161.
- Gateways: sólo un gateway de recepción.
- Condiciones: desfavorable debido a la multitud de edificios entre el gateway y el dispositivo emisor.

La tabla resultante del muestreo 6.1. A continuación, vemos la imagen con todas las localizaciones obtenidas (azules), junto a la localización de referencia (rojo) y la localización del gateway (verde) 6.5. Cabe mencionar que en la imagen sólo aparecen 14 muestras debido a que la muestra 1 y la muestra 12 son casi idénticas y apenas se distinguen en el mapa.

Muestra	Latitud	Longitud
1	43.36953	-8.41884
2	43.36792	-8.41950
3	43.36888	-8.41923
4	43.37163	-8.41518
5	43.37351	-8.42310
6	43.36929	-8.42752
7	43.36963	-8.42237
8	43.37030	-8.41898
9	43.37159	-8.42487
10	43.36875	-8.41528
11	43.36876	-8.41750
12	43.36954	-8.41885
13	43.36897	-8.42080
14	43.36994	-8.42019
15	43.37146	-8.42075

Tabla 6.1: Tabla muestreo localizaciones.

Con estos datos presentes, podemos concluir que, a pesar de encontrarse en la peor situación posible al sólo tener un gateway de referencia, el sistema obtuvo una localización bastante fiable ya que 9 de las 15 muestras se encontraban en un radio de 200 metros de la localización de referencia.

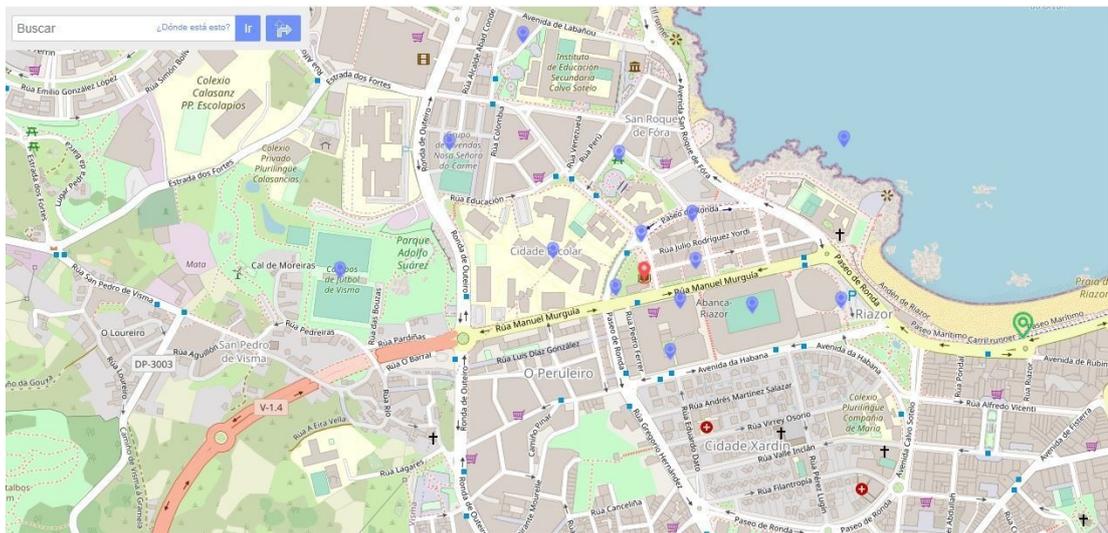


Figura 6.5: Mapeado del muestreo.

Planificación y evaluación de costes

EN este capítulo trataremos como se ha realizado la planificación, las incidencias surgidas a lo largo del proyecto y el coste final aproximado del mismo .

7.1 Planificación

Tal como lo mencionamos en el apartado de Metodología 1, hemos seguido una metodología Scrum sin paralelidad entre fases a causa de constar con un sólo miembro en el equipo del proyecto. En la imagen 7.1 podemos observar el diagrama de Gantt que hemos intentado cumplir a lo largo del proyecto. Esta planificación está formulada para 6 jornadas a la semana de 6 horas al día a lo largo de 9 semanas. Esto nos deja un total de 324 horas asignadas a este proyecto.

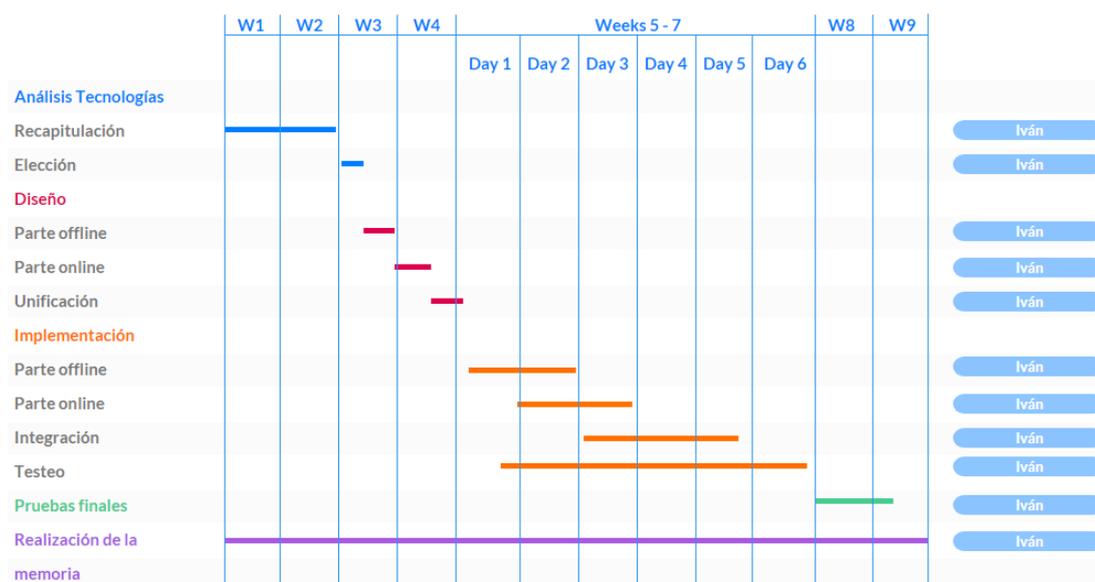


Figura 7.1: Diagrama de Gantt.

En el diagrama 7.1 se representa también los sprints. Estos se encuentran en el apartado de implementación (color naranja), repitiéndose durante 3 semanas correspondiendo cada una de esas 3 semanas para cada sprint.

7.1.1 Incidencias

A lo largo del proyecto hemos intentado seguir la planificación lo más estrictamente posible pero debido a situaciones imprevistas, problemas externos al equipo de desarrollo y algún error del mismo provocó retrasos importantes en la realización y finalización del proyecto. Las incidencias más importantes fueron:

- Mala adquisición del hardware: esta incidencia ocurre al comienzo del proyecto. Tras haber seleccionado el hardware en el cual se realizaría el proyecto, se procedió a su compra. La adquisición se realizó mediante el portal web Aliexpress a un vendedor poco conocido que nos brindaba un descuento que alcanzaba el 50% de los productos. Tras recibir los productos, realizamos un chequeo visual, que aprueban ya que coinciden con la descripción y con el producto deseado. A continuación, procedemos a configurarlos para usarlo en nuestro proyecto. Tras este punto y tras múltiples intentos y pruebas, nos percatamos que su funcionamiento es erróneo. Con esta información, nos ponemos en contacto con el vendedor que nos comunica que esa remesa de los productos habían salido defectuosa. Este defecto se localizaba en una pista de una capa interna de la PCB (dato extraído de la traducción del lenguaje chino proveniente del mensaje recibido por parte del vendedor). La solución propuesta por el vendedor fue el envío de nuevos productos pero sin asegurar la fiabilidad de estos y con un retraso en el envío de 30-60 días o la devolución del dinero. Escogimos la devolución del dinero (a pesar de perder el importe correspondiente al primer envío) para realizar una nueva compra a un vendedor de mayor confianza.
- Covid-19: esta situación debida a la aparición imprevista del Covid-19 y el posterior confinamiento y cuarentena decretada por el gobierno central afectó de manera negativa al proyecto. Este hecho provocó la ausencia del único miembro del equipo de desarrollo por un período de casi 4 meses.
- Escasez de cobertura en La Coruña: este hecho se produjo tras el confinamiento y afectó muy negativamente al proyecto tanto en la implementación como en las pruebas. La razón es porque antes del confinamiento había 7 gateways en La Coruña ciudad, llegando a 10 sumando los colindantes, pero tras el confinamiento se redujo esa cantidad a 2 gateways en ciudad y 5 sumando los gateways de la periferia. A esto se le sumaba la posibilidad de que algún gateway estuviera apagado llegando incluso a ocurrir que todos los gateways se encontraban apagados. En la imagen 7.2 observamos precisamente

como se encuentran todos los gateways apagados (captura del aplicativo TTN Mapper [38] que nos indica si el gateway se encuentra apagado con el color rojo, encendido con el color verde y si se encuentra encendido y disponible se nos muestra con el color azul), hecho que se llegó a prolongar en una ocasión durante varios días en la segunda quincena de agosto.

- Documentación errónea: este incidente fue provocado por el equipo del proyecto. Esto se debió a que cuando el equipo de desarrollo recopiló información técnica sobre el módulo WiFi LoRa 32 v2 Kit de Heltec accedió directamente a la ficha técnica del repositorio de Heltec sin indagar ni contrastar la información. Tras muchas horas y días de problemas y errores a la hora de integrar el programa Arduino con el módulo, el equipo decidió contrastar la información de la ficha técnica. En ese momento, el equipo se percató de que en el repositorio existía otra carpeta con casi el mismo nombre pero que en lugar de ser v2 era v2.1. Esta correspondía a una modificación del módulo v2 que tenía una importante diferencia de su predecesor en cuanto a nuestro proyecto se refería, la asignación de los pines. Resultó que el equipo de desarrollo había entrado por error en la carpeta v2.1 y obtenido el pdf de la ficha técnica de ahí, pdf que tenía el mismo nombre en ambas carpetas, hecho que retrasó el descubrimiento de la solución al problema.
- Accesibilidad en servicios web: uno de los problemas más recurrentes fue los cambios y caídas de servicios web como Collos, Lora Cloud o incluso la misma plataforma TTN. Para poder lidiar con ello, hubo que reajustar las tareas diarias, cambiar partes de la planificación e incluso añadir tareas adicionales como añadir una integración de reserva en la plataforma TTN para asegurar un correcto funcionamiento del sistema en caso de caída de Collos.

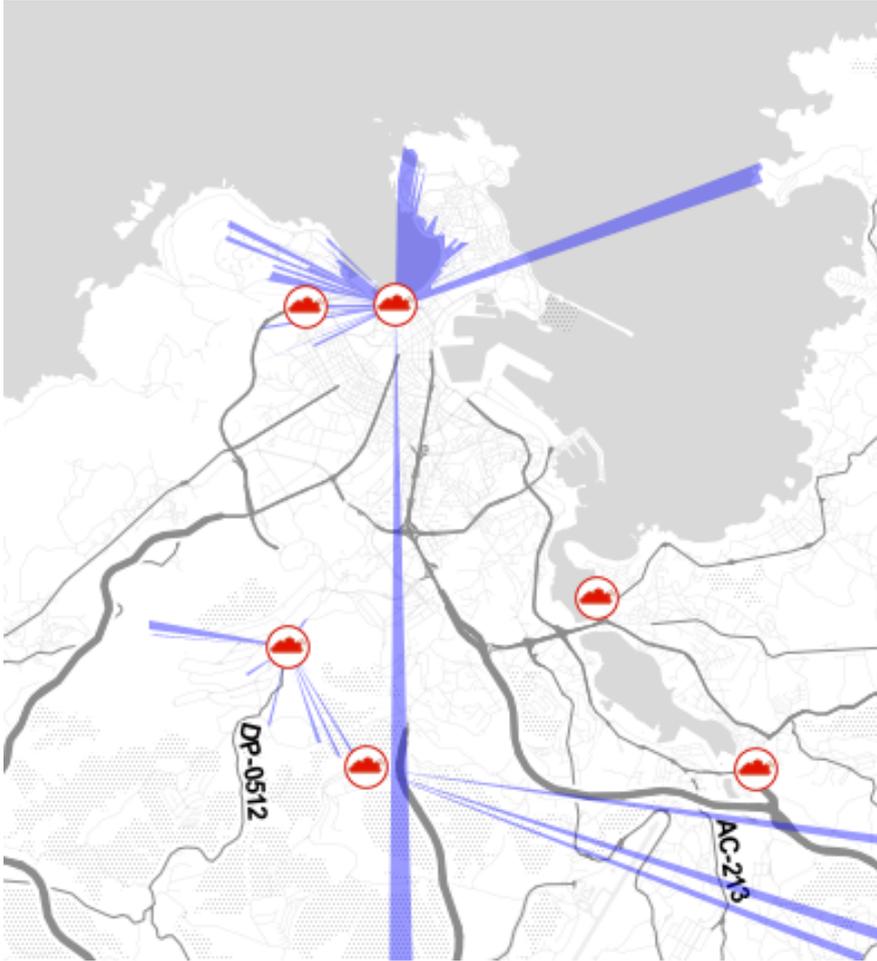


Figura 7.2: Mapa de la cobertura de la ciudad de La Coruña y su periferia.

7.2 Evaluación de costes

Ahora plasmaremos en varias tablas los costes aproximados derivados de la realización del proyecto. La primera de ellas es la tabla del coste de los recursos humanos utilizados 7.1. En este caso el coste es estimado.

	Coste/hora	Coste total (320h)	Total
Product Owner	50€	16.200€	
ScrumMaster	35€	11.340€	
Equipo de desarrollo	20€	6.480€	
-	-	-	34.200€

Tabla 7.1: Tabla coste recursos humanos.

La segunda tabla corresponde a los recursos materiales que fueron necesario para llevar a cabo el proyecto 7.2. Hay que aclarar dos puntos: gastos extras incluye todos los gastos derivados de las incidencias (la pérdida del importe del primer envío) y el material de oficina incluye tanto libretas y bolígrafos como impresiones y la pizarra magnética.

	Coste/hora	Coste total	Total
Ordenador portátil	-	695€	
Material de oficina	-	30€	
Módulo Heltec WiFi LoRa 32 v2 Kit	-	18,95€	
Protoboard y conectores	-	15€	
Módulo RFM95W	-	6,95€	
Módulo NodeMcu V3 Lua WiFi	-	3,15€	
Antena auxiliar	-	1,45€	
Pulsador	-	0,10€	
Plataforma TTN	0€	0€	
Plataforma IFTTT	0€	0€	
IDE Arduino	0€	0€	
Gastos extras	-	4,95€	
-	-	-	775,55€

Tabla 7.2: Tabla coste recursos materiales.

Por último, la tabla final 7.3 plasmará el coste total del proyecto sumando los recursos humanos y los recursos materiales. A mayores, la tabla diferenciará el coste total de dos maneras, el estimado que incluye todos los gastos descritos en las tablas anteriores y el real. El gasto real no cuenta con gastos de recursos humanos y se quitan varios puntos de los gastos materiales, los cuales son: el ordenador portátil y el material de oficina.

Coste	Total
Estimado	34.975,55€
Real	50,55€

Tabla 7.3: Tabla coste final.

Conclusiones

ESTE último capítulo de la memoria está destinado a realizar un repaso del proyecto, desde los requisitos cumplidos, las lecciones aprendidas y los pasos a seguir para mejorar el proyecto.

8.1 Conclusiones

Partiendo de un problema social de inseguridad e inquietud, comenzamos un proyecto para encontrar una solución que disminuyera dicho problema y sirviera de apoyo a los usuario para comunicarse y alertar tanto con a sus personas cercanas como a las fuerzas y cuerpos de seguridad del Estado de algún incidente o delito en su ubicación. Con esto en mente, analizamos las tecnologías inalámbricas más conocidas y nos decidimos por desarrollarlo según la tecnología IoT LoRa junto con su protocolo LoRaWAN. Acompañando a estas tecnologías utilizamos la plataforma The Things Network (TTN) como servidor y la plataforma IFTTT para llevar a cabo las acciones finales que ejecutará el sistema. Tras la finalización de este proyecto nos percatamos que obtuvimos un dispositivo funcional, compacto, de bajo coste y con una alta eficiencia energética pudiendo llegar a durar su batería hasta 10 años.

Además, comprobamos que conseguimos completar todos los requisitos acordados al comienzo del mismo, tanto los funcionales como los no funcionales. Realizando un pequeño repaso por ellos:

- Geolocalización sin GPS: hemos conseguido obtener una posición para el dispositivo sin necesidad de hardware adicional ni un mayor consumo de energía.
- Crear, modificar o eliminar secuencias: el dispositivo puede almacenar hasta 3 secuencias, las cuales se pueden modificar.
- Realizar una acción preconfigurada: tras la activación de la alerta y su recepción en TTN, el servicio IFTTT recibe la alerta y lleva a cabo la acción configurada y guardada

en su plataforma.

- Detener la alerta una vez activada: tras la activación de la alerta el dispositivo permite su detención introduciendo nuevamente la misma secuencia que la desencadenó.
- Diseño compacto: A pesar de que el dispositivo resultado del proyecto es un prototipo con un hardware diferente a la solución final, su tamaño ya es inferior al de una tarjeta bancaria.
- Bajo coste: a pesar de que el coste del prototipo es de 20€ (similar a los productos de la competencia), el dispositivo final con el hardware correspondiente tendría un coste de 10€. Este precio es bajo teniendo en cuenta que sería un precio de consumidor.
- Interfaz intuitiva:

A pesar de haber logrado el objetivo del proyecto, debemos hacer un análisis exhaustivo sobre su realización para extraer lecciones y experiencias que nos ayuden para proyectos futuros. Entre las lecciones obtenidas podemos destacar:

- Buena planificación: este es un punto clave en todo proyecto de cierto tamaño. Gracias a un buen asesoramiento, se realizó una planificación inicial que facilitó el desarrollo del proyecto y que a pesar de los retrasos surgidos se pudo mantener y amoldar a la situación.
- Imprevistos: en todo proyecto surgen imprevistos de todo tipo pero hay que intentar prevenirlos o anticiparse a ellos para que afecten lo menos posible a la consecución del proyecto. A pesar de esto, hay sucesos que no se pueden prevenir, por lo que se necesita maleabilidad para acondicionar la planificación y realizar los ajustes necesarios para amortiguarlo en la medida de lo posible.
- Falta de experiencia: durante el proyecto quedó presente que la falta de experiencia en el desarrollo y realización de proyectos de una envergadura mayor a una práctica de universidad puede provocar, en algunos casos, grandes retrasos o pérdidas económicas. Por ejemplo: una mala documentación del hardware puede provocar un cortocircuito de la placa y por consiguiente una nueva compra de material o un mal diseño provocado por la "metología primero programa y luego diseño" que provoca cambios importantes y pérdidas de tiempo.
- Concepto reciente: realizar un proyecto sobre un concepto o tecnología que surgió hace pocos años implica una falta de información y documentación importante. También implica que las herramientas para llevarlo a cabo son muy innovadoras y pueden contener bugs todavía sin descubrir. Para solventar esto, se necesita de una investigación

más profunda y un análisis mucho más exhaustivo para intentar proteger al proyecto de futuros errores. Esto supone un mayor gasto temporal al apartado de análisis en la planificación dejando un menor tiempo a repartir entre los demás puntos.

8.2 Líneas Futuras

A pesar de haber terminado el proyecto, el dispositivo resultado del mismo no está finalizado. Los objetivos que quedarían por hacer son:

- Miniaturizarlo: lo primero sería realizar una maquetación del diseño en una placa PCB. Luego ensamblar las partes a la placa y si todavía no estamos conformes con el tamaño, posteriormente miniaturizar ese diseño final.
- Cambiar la interfaz: convertir la interfaz de escritorio en una interfaz web accesible desde cualquier dispositivo con conexión a internet (ordenador, teléfono móvil, tablet ...) y actualizar el firmware del dispositivo en remoto desde la web mediante OTAA.
- Producción y distribución: Tras completar los dos puntos anteriores, realizaremos un estudio de mercado que nos determine si nuestro producto tendría buena acogida. De recibir una respuesta positiva procederíamos a producirlo en masa y posteriormente distribuirlo.

Apéndices

Material adicional

EN capítulo incluimos el material que creemos interesante para comprender mejor el proyecto.

A.1 Creación de una receta en IFTTT

Los pasos para crear una receta están descritos en el apartado de Aplicativo Online de Implementación 5. Aquí sólo mostraremos las capturas correspondientes.

Choose a service

Step 1 of 6



Figura A.1: IFTTT Paso 1.

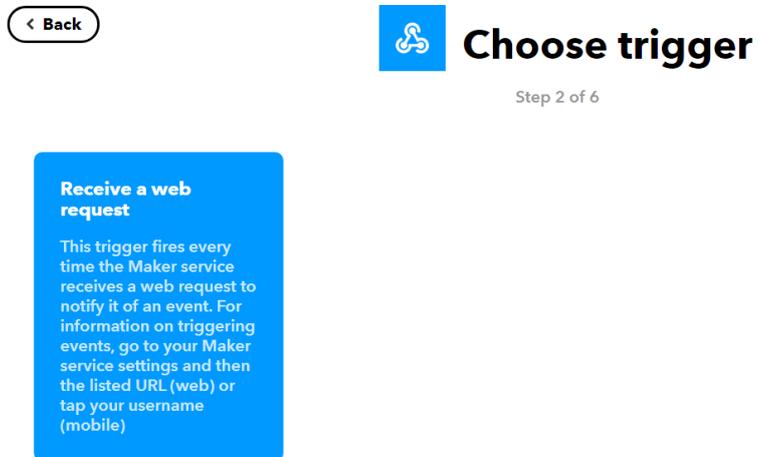


Figura A.2: IFTTT Paso 2.

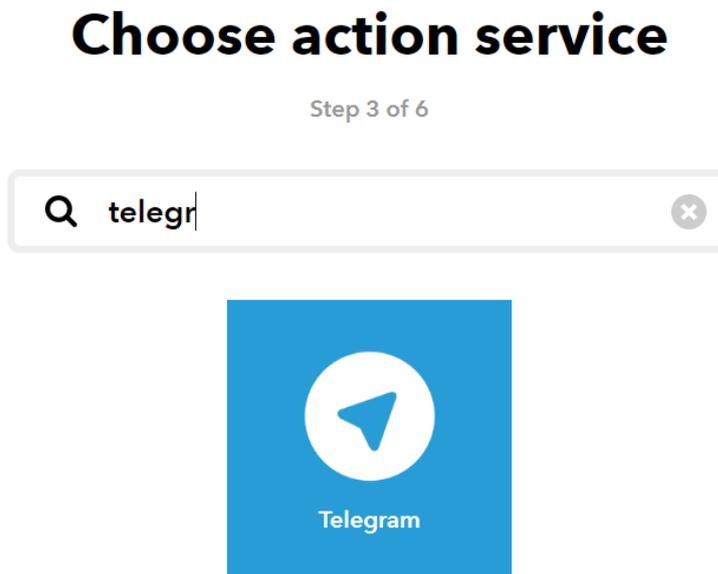


Figura A.3: IFTTT Paso 3.

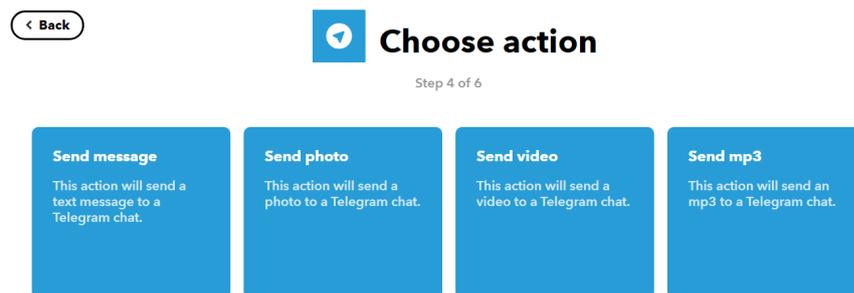


Figura A.4: IFTTT Paso 4.

Complete action fields

Step 5 of 6

Target chat

Private chat with @IFTTT ▼

Use the @IFTTT bot on Telegram to connect new groups or channels.

Message text

EventName

Secuencia: Value1 ,
[http://www.openstreetmap.org/?mlat= Value2 &mlon= Value3 #map=18](http://www.openstreetmap.org/?mlat=Value2&mlon=Value3#map=18)

Some HTML tags are supported: b, i, a, br, pre, code. Add ingredient

Include web page preview?

Enabled ▼

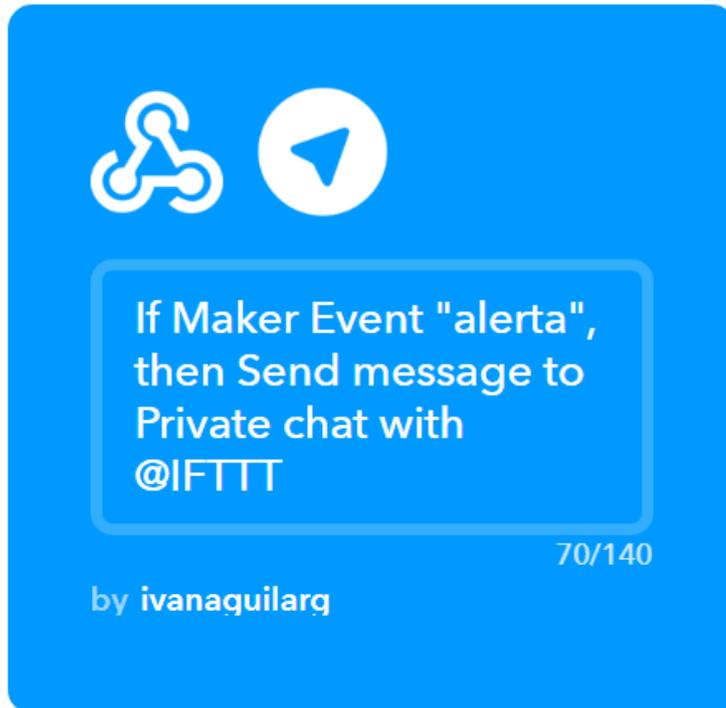
Telegram will check the first URL in the text to include a small preview of the page. Disabled by default.

Create action

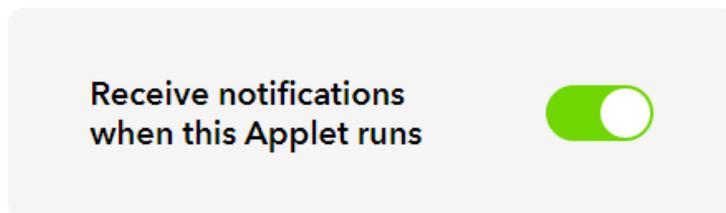
Figura A.5: IFTTT Paso 5.

Review and finish

Step 6 of 6



The image shows a blue rounded rectangle representing an IFTTT applet configuration. At the top left, there are two white icons: a circular icon with three nodes connected by lines, and a circular icon with a white triangle pointing right. Below these icons is a white-bordered box containing the text: "If Maker Event \"alerta\", then Send message to Private chat with @IFTTT". At the bottom right of the blue box, the text "70/140" is visible. At the bottom left, it says "by ivanaquilarg".



A light gray rounded rectangle containing a notification toggle. On the left, the text "Receive notifications when this Applet runs" is displayed. To the right of the text is a green toggle switch that is currently turned on.

Finish

Figura A.6: IFTTT Paso 6.

Lista de acrónimos

BLE Bluetooth Low Energy. 21, 31, 32

E/S Entrada/Salida. 10, 13

GPS Global Positioning System. 29, 31

Hz hercios. 18

IDE Integrated Development Environment. 13, 14

IEEE Institute of Electrical and Electronics Engineers. 21, 23

IoT Internet of Things. 2, 3, 7–9, 16–18, 29–31, 43, 75

LPWAN low-power wide-area network. 23, 24, 30

M2M machine to machine. 7

NVM Non-volatile memory. 9

SPOF single point of failure. 19

TTN The Things Network. 43, 45, 55, 57, 59, 71, 75

Glosario

chipset es el conjunto de circuitos integrados diseñados con base en la arquitectura de un procesador, permitiendo que ese tipo de procesadores funcionen en una placa base.. 18, 20, 21

comunidad maker es el concepto que engloba un grupo de personas que a través del open data y la tecnología comparten su propia experiencia con su comunidad para que puedan recrear sus productos y así contribuir en la innovación.. 26

EEPROM Es una memoria no volátil, la cual nos permite guardar datos y que no se borrarán aunque el sistema se quede sin energía.. 11

firmware es un programa informático que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo. Es el software que tiene directa interacción con el hardware, siendo así el encargado de controlarlo para ejecutar correctamente las instrucciones externas. Es uno de los tres principales pilares del diseño electrónico.. 30, 31

MEMORIA FLASH Es una memoria no volátil, la cual nos permite guardar datos y que no se borrarán aunque el sistema se quede sin energía. En comparativa, es más rápida que la memoria EEPROM.. 11

PCB Placa de circuito impreso constituida por caminos, pistas o buses de material conductor laminadas sobre una base no conductora.. 10

resistencia pull-down Son resistencias normales que establecen un estado LOW cuando el pin se encuentra en reposo. Esto evita los falsos estados que se producen por el ruido generado por los circuitos electrónicos.. 47

SRAM Memoria de acceso aleatorio y estático. Es donde el procesador guarda y manipula las variables del programa. Este tipo de memoria es volátil, es decir, cuando se le deja de aplicar un voltaje se borra.. 11

Bibliografía

- [1] Scrum. Home page. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.scrum.org/>
- [2] M. Gracia. (2019) Iot - internet of things. (accessed: 21.08.2020). [En línea]. Disponible en: <https://www2.deloitte.com/es/es/pages/technology/articles/IoT-internet-of-things.html>
- [3] K.-C. Ting-ChangChang. (2015) Resistance random access memory. Other authors: Tsung-MingTsai, Tian-JianChu, Simon M.Sze. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.sciencedirect.com/science/article/pii/S1369702115003843>
- [4] Adesto. (2016) Moneta ultra low power memory. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.adestotech.com/products/moneta-2/>
- [5] Arduino. (2005) Arduino web. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.arduino.cc/>
- [6] Processing. (2003) Processing foundation web. (accessed: 24.08.2020). [En línea]. Disponible en: <https://processing.org/>
- [7] Wiring. (2003) Wiring web. (accessed: 24.08.2020). [En línea]. Disponible en: <http://wiring.org.co/>
- [8] GNU. (2007) Gpl gnu web. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.gnu.org/licenses/gpl-3.0.html>
- [9] ——. (2007) Lgpl gnu web. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.gnu.org/licenses/lgpl-3.0.html>
- [10] C++. (2011) C++ reference. (accessed: 24.08.2020). [En línea]. Disponible en: <https://en.cppreference.com/w/cpp>

- [11] C. (2011) C reference. (accessed: 24.08.2020). [En línea]. Disponible en: <https://en.cppreference.com/w/c>
- [12] Wikipedia. Biblioteca estándar de c. (accessed: 24.08.2020). [En línea]. Disponible en: https://es.wikipedia.org/wiki/Biblioteca_est%C3%A1ndar_de_C
- [13] Arduino. Guía de referencia de arduino. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.arduino.cc/reference/es/>
- [14] ——. Guía de referencia de librerías arduino. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.arduino.cc/reference/en/libraries/>
- [15] O. Hab. Home page. (accessed: 2.09.2020). [En línea]. Disponible en: <https://www.openhab.org/>
- [16] H. Assistant. Home page. (accessed: 2.09.2020). [En línea]. Disponible en: <https://www.home-assistant.io/>
- [17] Node-Red. Home page. (accessed: 2.09.2020). [En línea]. Disponible en: <https://nodered.org/>
- [18] IFTTT. Home page. (accessed: 2.09.2020). [En línea]. Disponible en: <https://ifttt.com/>
- [19] Z. Alliance. Home page. (accessed: 24.08.2020). [En línea]. Disponible en: <https://zigbeealliance.org/es/>
- [20] Z-Wave. Home page. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.z-wave.com/>
- [21] S. Labs. Home page. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.silabs.com/>
- [22] Bluetooth. Home page. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.bluetooth.com/>
- [23] W.-F. Alliance. Home page. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.wi-fi.org/>
- [24] I. SA. Ieee 802.11b-1999. (accessed: 24.08.2020). [En línea]. Disponible en: https://standards.ieee.org/standard/802_11b-1999.html
- [25] Weightless. Home page. (accessed: 24.08.2020). [En línea]. Disponible en: <http://www.weightless.org/>

- [26] 3GPP. Home page. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.3gpp.org/>
- [27] SigFox. Home page. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.sigfox.es/>
- [28] Semtech. Home page. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.semtech.com/>
- [29] L. Alliance. Home page. (accessed: 24.08.2020). [En línea]. Disponible en: <https://lora-alliance.org/>
- [30] TTN. Record lora. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.thethingsnetwork.org/article/lorawan-distance-world-record>
- [31] D. Matter. Home page. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.digitalmatter.com>
- [32] ——. Guppy lorawan. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.digitalmatter.com/devices/guppy-lorawan/>
- [33] MOKOSmart. Home page. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.mokosmart.com/>
- [34] ——. Lw004-pb. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.mokosmart.com/es/lorawan-button-lw004-pb/>
- [35] ——. Lw004-ct. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.mokosmart.com/es/lw004-lorawan-contact-tracing-wearables-button/>
- [36] TTN. Home page. (accessed: 24.08.2020). [En línea]. Disponible en: <https://www.thethingsnetwork.org/>
- [37] SEMTECH. Home page. (accessed: 2.09.2020). [En línea]. Disponible en: <https://www.loracloud.com/>
- [38] T. Mapper. Home page. (accessed: 31.08.2020). [En línea]. Disponible en: <https://ttnmapper.org/>

